RAPPORT DE TER sur PRELUDE-IDS

Clément LORVAO, Dado KONATE, Guillaume LEHMANN (lehmann@free.fr)

9 avril 2004

Page de note

Table des matières

1 Note de mise à jour **6**

2 Introduction **7**

3 Quelques notions de sécurité **8**
3.1 Présentation 8
3.2 Où interviennent les IDS dans une politique de sécurité ? 9

4 Les IDS **10**
4.1 Généralités sur les systèmes de détection d'intrusion 10
4.1.1 Introduction 10
4.1.2 Critères de classification des IDS 10
4.1.3 Les différents types d'IDS 11
4.2 Points forts de cette méthode de protection 13
4.2.1 Une surveillance continue et détaillée 13
4.2.2 Contrôle du payload 13
4.2.3 Modularité de l'architecture 14
4.2.4 Réponse active 14
4.2.5 Les HIDS complètent l'analyse des NIDS 14
4.2.6 Combinaison avec d'autres outils 15
4.2.7 Bonne protection des IDS eux-mêmes 15
4.3 Points faibles de cette méthode de protection 15
4.3.1 Besoin de connaissances en sécurité 15
4.3.2 Problème de positionnement des sondes 15
4.3.3 Vulnérabilités des sondes NIDS 16
4.3.4 Un DoS explose les fichiers de logs 16
4.3.5 Réponse active non-contrôlée 16
4.3.6 Problèmes de IPv4 17
4.3.7 Problèmes intrinsèques à la plateforme 17
4.3.8 Gestion de plusieurs managers très limitée 17
4.3.9 Coût matériel parfois important 17
4.4 Méthodes de contournement des IDS 17
4.4.1 Quelques techniques 18
4.4.2 Exemples d'attaques 18
4.5 Une alternative aux IDS 19
4.5.1 La tolérance d'intrusion 19
4.6 Conclusion 20

5 Prelude-IDS et ses compères **21**
5.1 Historique 21
5.2 Caractéristiques 21
5.2.1 Description 21
5.2.2 Architecture 21
5.2.3 Composants 22
5.3 Prelude-IDS et Snort 23
5.3.1 Projet 23
5.3.2 Moteur d'analyse et banque de signatures 23
5.3.3 La remontée d'alertes 24
5.3.4 Infos disponibles dans les alertes 24

5.3.5 Les frontends . . . 24
5.3.6 Intégration d'outils externes . . . 30
5.3.7 Facilité de configuration et bonne documentation . . . 30
5.4 Prelude-IDS et LogCheck . . . 30
5.4.1 Présentation de LogCheck . . . 31
5.4.2 Fonctionnement de LogCheck . . . 31
5.4.3 Installation de LogCheck . . . 31
5.4.4 Comparaison entre Prelude-IDS et Logcheck . . . 31
5.5 Prelude-IDS et Nessus . . . 31
5.5.1 Nessus, généralités . . . 31
5.5.2 Comparaison entre Nessus et Prelude-IDS . . . 32
5.5.3 Conclusion . . . 33

6 Prelude-IDS et ses compères **33**
6.1 Prelude-IDS, Honeyd et Systrace . . . 33
6.1.1 Honeyd . . . 33
6.1.2 Systrace . . . 34

7 Utilisation de Prelude (positionnement dans un réseau) **36**
7.1 Suivi des attaques . . . 38
7.2 Surveillance simple . . . 40
7.3 Surveillance ciblée . . . 42
7.4 Surveillance multi-sites, multi-responsabilités . . . 44

8 Conclusion **46**

9 Bibliographie **47**
9.1 Installation et configuration de Prelude-IDS . . . 47
9.2 Aide sur les Systèmes de Détection d'Intrusion . . . 47
9.3 Bibliographie pour intégrer les règles de Snort à Prelude-NIDS . . . 47
9.4 Projets liés de Prelude-IDS . . . 47

10 Licence **49**

A Insertion des règles de Snort dans Prelude-NIDS **50**
A.1 Intégration des règles . . . 50
A.2 Le code source du script convert_ruleset . . . 50
A.3 Le code source du script convert_ruleset commenté . . . 51

B Installation de Honeyd avec Prelude-IDS **53**

C Installation de Nessus et corrélation avec Prelude-IDS **55**
C.1 Installation de Nessus . . . 55
C.2 Intégration de Nessus dans Prelude . . . 56

D Manuel d'installation de Prelude-lml/NIDS/manager, et des frontends perl et php **59**
D.1 Introduction . . . 59
D.2 Paquetages nécessaires . . . 59
D.3 Installation de paquetages au préalable . . . 59
D.3.1 Paquetages nécessaires à Prelude-IDS . . . 59
D.3.2 Installation de ces paquetages . . . 59
D.4 Installation . . . 60

D.4.1 Installation de libprelude . . . 60
D.4.2 Installation du manager . . . 60
D.4.3 Installation de la sonde réseau (nids) . . . 60
D.4.4 Installation de la sonde hôte (lml) . . . 60
D.5 Configuration . . . 60
D.5.1 Configuration de MySQL . . . 60
D.5.2 Configuration du manager . . . 61
D.5.3 Configuration de la sonde réseau (nids) . . . 61
D.5.4 Configuration de la sonde hôte (lml) . . . 61
D.6 Lancement de l'écoute . . . 62
D.7 Installation et configuration du prelude-php-frontend . . . 62
D.7.1 Installation . . . 62
D.7.2 Configuration . . . 63
D.8 Installation et configuration du prelude-perl-frontend . . . 63
D.8.1 Installation préalable de paquetages . . . 63
D.8.2 Installation de prelude-perl-frontend . . . 64
D.8.3 Configuration d'Apache . . . 64
D.8.4 Configuration de prelude-perl-frontend . . . 65
D.9 Liens . . . 65
.1 Liste de paquetages installés sur le système . . . 66
.2 Fichier httpd.conf . . . 71
.3 Fichier prelude-manager.conf . . . 95
.4 Fichier prelude-nids.conf . . . 97
.5 Fichier prelude-lml.conf . . . 100
.6 Fichier config.pl . . . 101
.7 Fichier config.php . . . 102

1 Note de mise à jour

Dans ce rapport, il est expliqué que le perl-frontend de Prelude ne permet pas de supprimer des alertes de la bases de données. C’est une erreur, car en accès *administrateur*, cette fonctionnalité est offerte.

2 Introduction

Ce rapport présente le Travail d'Etude et de Recherche (TER) réalisé par Guillaume Lehmann, Dado Konate et Clément Lorvao. Ce travail a été réalisé sur 13 semaines à raison de 2 jours par semaine dans le cadre du DESS STRI. Nous étions encadrés par Mr. Philippe Latu.

L'objectif était d'étudier les possibilités de l'outil Prelude-IDS, puis de le comparer à la solution Snort/Logcheck/ACID. Cette dernière solution est présente sur le réseau de l'Institut Universitaire de Technologie de Toulouse, et nous devions la remplacer par Prelude-IDS si le nouvel outil répondait mieux aux besoins sécurité.
Pour réaliser cette étude, nous devions travailler sur 6 sondes (essentiellement NIDS) et 1 manager. Nous utilisions le système d'exploitation Linux, et travaillions sur une distribution Debian/Woody. L'absence de paquetages officiels de Prelude-IDS sous Debian nous a contraint à travailler directement à partir des sources officielles de l'outil.
Nous disposions aussi d'autres machines plus puissantes connectées à Internet.

Le rapport est en fait le résultat d'une étude beaucoup plus large. En effet, il y est question des avantages et inconvénients d'une solution par rapport à l'autre, mais on évoque aussi la mise en place de ces outils. Sont aussi traités l'installation et la configuration d'outils comme Nessus ou Honeyd qui peuvent interagir avec Prelude. Puis de manière plus générale, nous évoquons les apports des IDS à la sécurité informatique. En bref, ce rapport couvre la théorie et la pratique des IDS et des outils de sécurité, avec Prelude-IDS comme fil conducteur tout au long de l'étude.

3 Quelques notions de sécurité

3.1 Présentation

Avant de présenter l'étude de l'outil Prelude-IDS, ou des Systèmes de Détection d'Intrusions de manière plus générale, nous allons présenter notre vision de la sécurité.
Nous pouvons découper le domaine de la sécurité de la façon suivante :
- *La sécurité logicielle* qui gère la sécurité du Système d'Information au niveau logiciel, et qui intègre aussi des protections logicielles comme les antivirus.
- *La sécurité du personnel* qui comprend la formation et la sensibilisation des personnes utilisant ou travaillant avec le Système d'Information.
- *La sécurité physique* qui regroupe la politique d'accès aux bâtiments, la politique d'accès au matériel informatique, et les règles de sécurité pour la protection des équipements réseaux actifs et passifs.
- *La sécurité procédurale* définit les procédures et les règles d'utilisation du Système d'Information.
- *La sécurité réseau* où sont gérés l'architecture physique et logique du réseau, les politiques d'accès aux différents services, la gestion des flux d'informations sur les réseaux, et surtout les points de contrôles et de surveillance du réseau.
- *La veille technologique* qui est souvent oubliée et qui permet de faire évoluer la sécurité au cours du temps afin de maintenir un niveau de protection du système d'information suffisant.

Après avoir vu le découpage organisationnel de la sécurité, intéressons-nous aux objectifs. Une bonne politique de sécurité doit préserver les aspects de :
- *Disponibilité* : c'est-à-dire fournir l'accès à l'information pour que les utilisateurs autorisés puissent la lire ou la modifier. Ou encore faire en sorte qu'aucune personne ne puisse empêcher les utilisateurs autorisés d'accéder à l'information.
- *Confidentialité* : c'est-à-dire empêcher les utilisateurs de lire une information confidentielle (sauf s'ils y sont autorisés). Ou encore, empêcher les utilisateurs autorisés à lire une information, de la divulguer à d'autres utilisateurs (sauf autorisation).
- *Intégrité* : c'est-à-dire empêcher une modification (création, mise à jour, ou destruction) indue de l'information. Ou encore faire en sorte qu'aucun utilisateur ne puisse empêcher la modification légitime de l'information.

Il existe des sous-définitions comme *l'intimité* qui est un cas particulier de *la confidentialité,* ou encore *la non-répudiation* ou *la pérennité*...

Toute entrave à la sécurité peut être modélisée de la façon suivante :

```
CAUSE                              ETAT                                SERVICE
Faute ------------------------ ->Erreur ---------------------- ->Défaillance
```

- FAUTE : Cause adjugée ou supposée d'une erreur.
- ERREUR : Au moins une partie du système suit un comportement erroné, susceptible d'entraîner une défaillance.
- DEFAILLANCE : Le service délivré par le système dévie du service spécifié. L'incident de sécurité est arrivé à son terme.

Afin de garantir une sécurité suffisante, il faut que toute attaque soit bloquée pendant l'une des 3 phases (et surtout avant la fin de la troisième). Généralement, plus on se rapproche de la défaillance, plus le problème sera difficile et long à résoudre ... et que l'on se rapproche de la réussite de l'attaque.

3.2 Où interviennent les IDS dans une politique de sécurité ?

Suivant le découpage précédent, nous pouvons dire que la protection des IDS intervient dès l'apparition de la faute (ou début d'attaque). Certains IDS peuvent répondre à cette attaque automatiquement (contre-attaque directe, changement du routage, ping hole, ...) ce qui permet de bloquer immédiatement l'attaque. Cependant, la plupart des IDS ne font que remonter une alerte, et la réussite d'une attaque dépendra alors du temps de réaction du responsable sécurité.

Rapidement, nous pouvons définir au-moins 2 types d'analyses réalisées par les IDS (cela sera explicité dans le détail dans la suite de ce rapport) :
– par étude comportementale (la secrétaire se met tout-à-coup à programmer en assembleur),
– par étude de correspondance à un scénario préalablement défini.
Que ce soit la première ou la deuxième solution, un IDS ne protège pas plus particulièrement l'intégrité des données, la disponibilité des services, ou encore la confidentialité des informations. Un aspect de la sécurité n'est pas plus prioritaire qu'un autre.

Au niveau organisationnel, nous pouvons intégrer les IDS au sein de :

la sécurité réseau : puisque l'IDS va protéger des attaques venant ou transitant sur le réseau ;

la veille technologique : puisque la mise-à-jour des règles, et la surveillance des alertes sont liées directement à l'efficacité de l'IDS.

4 Les IDS

4.1 Généralités sur les systèmes de détection d'intrusion

4.1.1 Introduction

La détection d'intrusion a pour objectif de détecter toute violation de la politique de sécurité sur un système informatique. Elle permet ainsi de signaler les attaques (en temps réel ou en différé) portant atteinte à la sécurité de ce système.

- Par sécurité du système, nous considérons l'intégrité, la confidentialité et la disponibilité du système et des données.
- Par attaque, nous ne considérons pas seulement les intrusions ou tentatives d'intrusion mais aussi d'autres actions telles que les scans, dénis de service, utilisations non autorisées de systèmes/services, mauvaises utilisations de systèmes/services...

Pour mettre en oeuvre ce concept de détection d'intrusion, des outils spécifiques sont nécessaires : les IDS ou systèmes de détection d'intrusions. Ils vont permettre de collecter de façon automatisée les données représentant l'activité des systèmes(serveurs, applications, systèmes, réseaux), de les analyser et d'avertir les administrateurs en cas de détection de signes d'attaques.

4.1.2 Critères de classification des IDS

Méthodes d'analyse
Le premier critère de classification des IDS reste la méthode d'analyse. Deux approches sont possibles :

L'approche par scénario : Cette approche consiste à rechercher dans l'activité de l'élément surveillé les empreintes (ou signatures) d'attaques connues. Ce type d'IDS est purement réactif ; il ne peut détecter que les attaques dont il possède la signature. De ce fait, il nécessite des mises à jour fréquentes. De plus, l'efficacité de ce système de détection dépend fortement de la précision de sa base de signature. C'est pourquoi ces systèmes sont contournés par les pirates qui utilisent des techniques dites "d'évasion" qui consistent à maquiller les attaques utilisées. Ces techniques tendent à faire varier les signatures des attaques qui ainsi ne sont plus reconnues par l'IDS.

L'approche comportementale : Elle consiste à détecter des anomalies. La mise en oeuvre comprend toujours une phase d'apprentissage au cours de laquelle les IDS vont "découvrir" le fonctionnement "normal" des éléments surveillés. Ils sont ainsi en mesure de signaler les divergences par rapport au fonctionnement de référence. Les modèles comportementaux peuvent être élaborés à partir d'analyses statistiques. Ils présentent l'avantage de détecter des nouveaux types d'attaques.Cependant, de fréquents ajustements sont nécessaires afin de faire évoluer le modèle de référence de sorte qu'il reflète l'activité normale des utilisateurs et réduire le nombre de fausses alertes générées.

Chacune de ces deux approches peut conduire à des faux-positifs (détection d'attaque en absence d'attaque) ou à des faux-négatifs (absence de détection en présence d'attaque).

Autres critères
Parmi les autres critères de classification existants, nous pouvons citer entre autres :

- les sources de données à analyser (réseau/système/application),
- le comportement de l'IDS après intrusion (passif/actif),
- la fréquence d'utilisation (périodique/continue).

4.1.3 Les différents types d'IDS

Un IDS a pour fonction d'analyser en temps réel ou différé les évènements en provenance des différents systèmes, de détecter et de prévenir en cas d'attaque. Les buts sont nombreux :

– collecter des informations sur les intrusions,
– gestion centralisée des alertes,
– effectuer un premier diagnostic sur la nature de l'attaque permettant une réponse rapide et efficace,
– réagir activement à l'attaque pour la ralentir ou la stopper.

Les systèmes de détection d'intrusion ou IDS peuvent se classer selon trois catégories majeures selon qu'ils s'attachent à surveiller :

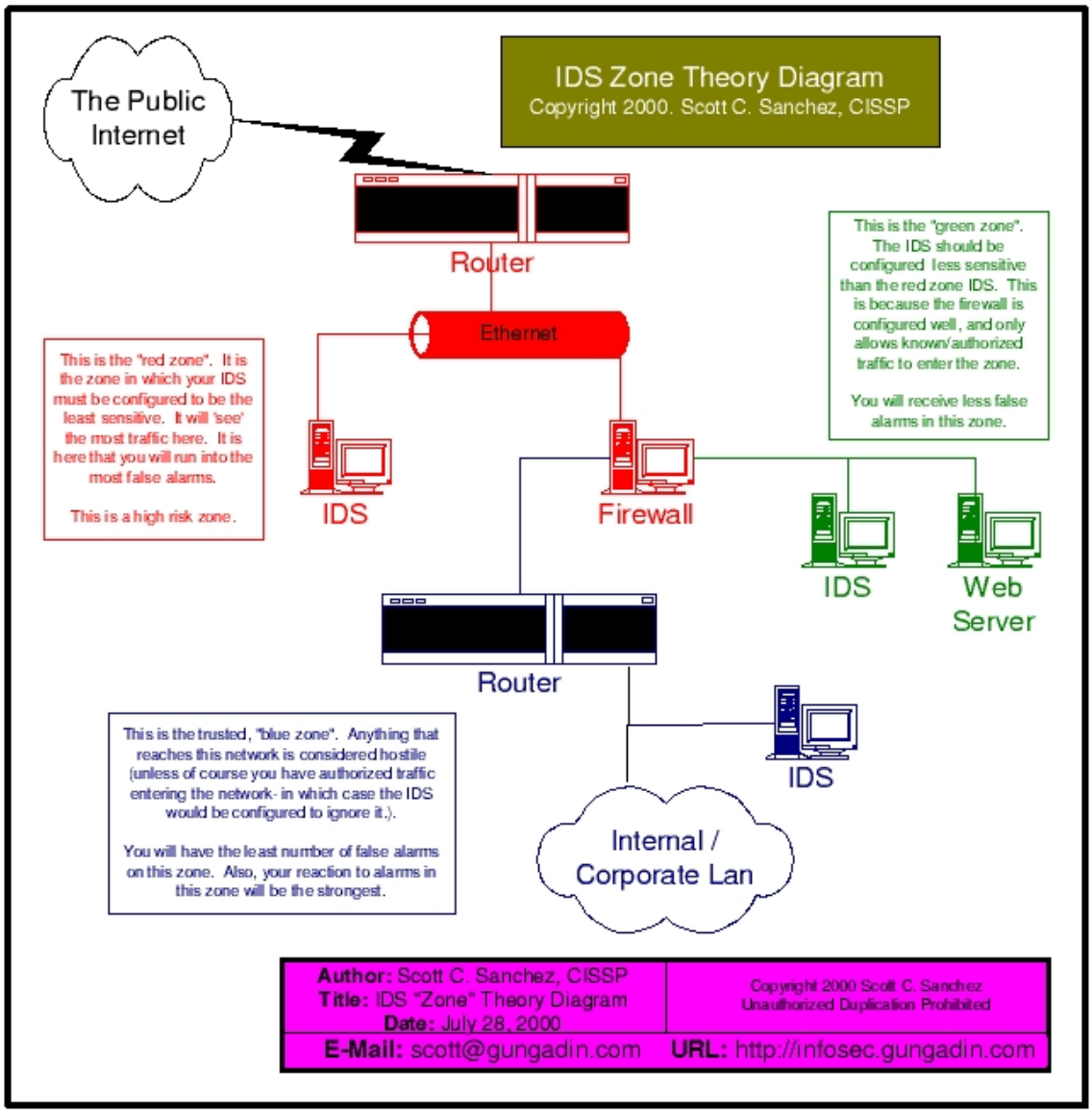
– le trafic réseau : on parle d'IDS réseau ou NIDS(Network based IDS)
– l'activité des machines : on parle d'IDS Système ou HIDS(Host based IDS)
– une application particulière sur la machine : on parle d'IDS Application (Application based IDS). Contrairement aux deux IDS précédents, ils sont rares. Nous ne les traiterons donc pas.

Les NIDS
Ces outils analysent le trafic réseau ; ils comportent généralement une sonde qui "écoute" sur le segment de réseau à surveiller et un moteur qui réalise l'analyse du trafic afin de détecter les signatures d'attaques ou les divergences face au modèle de référence. Les IDS Réseaux à base de signatures sont confrontés actuellement à deux problèmes majeurs qui sont : l'utilisation grandissante du cryptage, et des réseaux commutés. En effet, il est d'une part plus difficile " d'écouter " sur les réseaux commutés et le cryptage rend l'analyse du contenu des paquets presque impossible. La plupart des NIDS sont aussi dits IDS inline car ils analysent le flux en temps réel. Pour cette raison, la question des performances est très importante. De tels IDS doivent être de plus en plus performants afin d'analyser les volumes de données de plus en plus importants pouvant transiter sur les réseaux.

Les HIDS
Les IDS Systèmes analysent quant à eux le fonctionnement ou l'état des machines sur lesquelles ils sont installés afin de détecter les attaques. Pour cela ils auront pour mission d'analyser les journaux systèmes, de contrôler l'accès aux appels systèmes, de vérifier l'intégrité des systèmes de fichiers ... Ils sont très dépendants du système sur lequel ils sont installés. Il faut donc des outils spécifiques en fonction des systèmes déployés. Ces IDS peuvent s'appuyer sur des fonctionnalités d'audit propres ou non au système d'exploitation, pour en vérifier l'intégrité, et générer des alertes. Il faut cependant noter qu'ils sont incapables de détecter les attaques exploitant les faiblesses de la pile IP du système, typiquement les Dénis de service comme SYN FLOOD ou autre.

Les IDS hybrides (NIDS+HIDS)
Les IDS hybrides rassemblent les caractéristiques de plusieurs IDS différents. En pratique, on ne retrouve que la combinaison de NIDS et HIDS. Ils permettent, en un seul outil, de surveiller le réseaux et les terminaux. Les sondes sont placées en des points stratégiques, et agissent comme NIDS et/ou HIDS suivant leurs emplacements. Toutes ces sondes remontent alors les alertes à une machine qui va centraliser le tout, et agréger/lier les informations d'origines multiples.

Placement des IDS
Le placement des IDS va dépendre de la politique de sécurité menée. Mais il serait intéressant de placer des IDS :
– dans la zone démilitarisée (attaques contre les systèmes publics),
– dans le (ou les) réseau privé (intrusions vers ou depuis le réseau interne),
– sur la patte extérieure du firewall (détection de signes d'attaques parmi tout le trafic entrant et sortant, avant que n'importe quelle protection intervienne).


IDS Zone Theory Diagram
Copyright 2000. Scott C. Sanchez, CISSP
The Public Internet
Router
Ethernet
This is the "red zone". It is the zone in which your IDS must be configured to be the least sensitive. It will 'see' the most traffic here. It is here that you will run into the most false alarms.
This is a high risk zone.
IDS
Firewall
This is the "green zone". The IDS should be configured less sensitive than the red zone IDS. This is because the firewall is configured well, and only allows known/authorized traffic to enter the zone.
You will receive less false alarms in this zone.
IDS
Web Server
Router
This is the trusted, "blue zone". Anything that reaches this network is considered hostile (unless of course you have authorized traffic entering the network- in which case the IDS would be configured to ignore it.).
You will have the least number of false alarms on this zone. Also, your reaction to alarms in this zone will be the strongest.
IDS
Internal / Corporate Lan
Author: Scott C. Sanchez, CISSP
Title: IDS "Zone" Theory Diagram
Date: July 28, 2000
E-Mail: scott@gungadin.com
Copyright 2000 Scott C. Sanchez
Unauthorized Duplication Prohibited
URL: http://infosec.gungadin.com

Il est important de bien définir les zones sensibles du système d'information, ainsi que les zones les plus attractives pour un pirate. Suivant que nous voulons faire du suivi, un audit, ou encore une simple détection (unique) des attaques, nous placerons plus ou moins de sondes sur le réseau.

Il faut aussi voir qu'au-delà de l'architecture du réseau, il faut prendre en compte l'organisation de la sécurité existante :
– recherche-t-on une administration centralisée ?
– quel est l'existant organisationnel de la surveillance du réseau ?
– quels sont les compétences et les moyens en internes pour gérer les IDS ?

Des cas concrets seront donnés à travers les explications sur le placement des sondes Prelude-IDS. Il faut cependant retenir que plusieurs cas sont envisageables, et une réflexion sur le réseau est à mener en amont.

L'évolution du marché

Il existe aujourdui de nombreux produits dont la complexité de mise en oeuvre et le degré d'intégration sont très divers. Les outils strictement basés sur des modèles comportementaux sont actuellement en perte de vitesse. Mais ils sont de plus en plus intégrés à des IDS initialement basés sur une bibliothèque de signatures, étant donné leur complémentarité. Les IDS systèmes sont un peu en retrait face aux IDS réseaux. L'arrivée des IDS hybrides, qui apportent une sécurité moins parcellaire dans la protection du système d'information, pourrait remettre en question l'état du marché des IDS.

Les critères de choix

Les systèmes de détection d'intrusion sont devenus indispensables lors de la mise en place d'une infrastructure de sécurité opérationnelle. Ils s'intègrent donc toujours dans un contexte et dans une architecture imposants des contraintes très diverses. Certains critères (toujours en accord avec le contexte de l'étude) peuvent être dégagés :

Fiabilité : Les alertes générées doivent être justifiées et aucune intrusion ne doit pouvoir lui échapper.

Réactivité : Un IDS doit être capable de détecter les nouveaux types d'attaques le plus rapidement possible ; pour cela il doit rester constamment à jour. Des capacités de mise à jour automatique sont pour ainsi dire indispensables.

Facilité de mise en oeuvre et adaptabilité : Un IDS doit être facile à mettre en oeuvre et surtout s'adapter au contexte dans lequel il doit opérer ; il est inutile d'avoir un IDS émettant des alertes en moins de 10 secondes si les ressources nécessaires à une réaction ne sont pas disponibles pour agir dans les mêmes contraintes de temps.

Performance : la mise en place d'un IDS ne doit en aucun cas affecter les performance des systèmes surveillés. De plus, il faut toujours avoir la certitude que l'IDS a la capacité de traiter toute l'information à sa disposition (par exemple un IDS réseau doit être capable de traiter l'ensemble du flux pouvant se présenter à un instant donné sans jamais supprimer de paquets) car dans le cas contraire il devient trivial de masquer les attaques en augmentant la quantité d'information.

4.2 Points forts de cette méthode de protection

Mettre en place un réseau est assez facile. Des technologies comme Ethernet pour les réseaux locaux sont maintenant stables. Le soucis aujourd'hui est d'aller plus loin que la mise en place d'un réseau qui répond à nos besoins. Nous voulons contrôler ce qui s'y passe. Cela met en avant la QoS pour gérer au mieux son réseau, ou la sécurité pour protéger au mieux ce qui nous appartient.

4.2.1 Une surveillance continue et détaillée

Dans cette optique, nous nous interéssons aux flux valides, mais aussi au flux non-valides qui transitent sur notre réseau dont nous avons la responsabilité. Comment savoir si les règles d'un firewall sont valides ? Comment savoir le nombre d'attaques subies au cours de la dernière semaine ? Comment différencier une surcharge normale du réseau d'une attaque par DoS ?

Les IDS vont répondre à ces questions. Ce sont des sondes en mode promiscuité. Ils peuvent donc analyser tout le traffic (dans le même domaine de collision), et relever des attaques, alors même qu'ils n'en sont pas la cible directe. Biensûr, nous évoquons ici le fonctionnement des NIDS. Les HIDS vont au contraire établir une suveillance unique du système sur lequel ils sont installés.
De plus, toutes les alertes sont stockées soit dans un fichier, soit dans un base de données, ce qui permet de concevoir un historique, et détablir des liens entre différentes attaques. Ainsi, le responsable sécurité n'a pas besoin de surveiller en permanence pour être au courant de ce qui se passe sur le réseau. Une attaque de nuit ne passera plus inaperçue.
Tous les IDS renvoient de nombreuses informations avec une alerte. Le type supposé d'attaque, la source, la destination, ... Tout cela permet un bonne compréhension d'un incident sécurité, et en cas de faux-positif, de le détecter rapidement.

4.2.2 Contrôle du payload

Un autre point important dans la sécurité : nous avons maintenant des outils de filtrage très intéressants qui nous permettent de faire du contrôle par protocole (icmp, tcp, udp), par adresse IP, jusqu'à du suivi de

connexion (couches 3 et 4). Même si cela écarte la plupart des attaques, cela est insuffisant pour se protéger des attaques passant par des flux autorisés. Si cela est assez marginal, car difficile à mettre en place, l'ouverture de l'informatique au grand public et l'augmentation de ce type de connaissances font qu'il faudra un jour savoir s'en protéger efficacement. Nous pouvons évoquer les firewalls au niveau applicatif, mais de manière générale, la technique n'est pas au point. C'est ici qu'interviennent les IDS. Les IDS contrôlent tout le traffic, quel que soit le service (pop3, www, nntp, ...). C'est le contenu des trames qui est surveillé. C'est tout de même à différencer des antivirus qui étudient le code. Ici, les NIDS étudient la possible nuisance de données et ne contrôlent pas le contenu des fichiers... quoique, Prelude-IDS contient une règle pour parer l'attaque du vers qui attaquait les SGBD MSSQL.
De plus, certains IDS comme Prelude-IDS archivent le payload des trames suspectes, ce qui permet à l'administrateur d'étudier le problème. Nous avons donc une solution efficace et bon marché de contrôler les payloads.

4.2.3 Modularité de l'architecture

Il y a plusieurs solutions pour le positionnement de sondes réseaux.
Il peut être intéressant de positionner les sondes pour étudier l'efficacité des protections mises en place. Par exemple dans un réseau se cachant derrière un firewall, nous mettrons une sonde côté extérieur du firewall, et une autre côté intérieur du firewall. La première sonde permet de détecter les tentatives d'attaques dirigées contre le réseau surveillé. La seconde sonde va remonter les attaques (préalablement détectées par la première sonde) qui ont réussi à passer le firewall. On peut ainsi suivre une attaque sur un réseau, voir si elle arrive jusqu'à sa victime, en suivant quel parcours, ...
Il est aussi intéressant de définir des périmètres de surveillance d'une sonde. Ce sera en général suivant un domaine de collision, ou sur des entrées uniques vers plusieurs domaines de collision (par exemple à l'entrée d'un commutateur). Par cette méthode, nous réduisons le nombre de sondes, car il n'y a pas de doublons dans la surveillance d'une partie du réseau. Une alerte n'est remontée qu'une seule fois ce qui allège d'autant l'administration des IDS. Et pour finir, le fait de placer les sondes après les protections est plus logique, car le but premier des IDS est d'étudier les intrusions malgré les protections.

Ensuite à chacun de créer ses propres architectures suivant ses besoins et son imagination. On voit que la modularité de mise en place permet plusieurs types de surveillances. Même si la surveillance des sondes est limitée à un domaine de collision, l'architecture client-serveur ne l'est pas puisque cette dernière se base sur un repérage par adresse IP.

4.2.4 Réponse active

Prelude-IDS est un exemple d'IDS actif. Lorsqu'il détecte une attaque, il contre-attaque. Par exemple, s'il détecte un ping flood, il va effectuer une attaque par IP Fragment sur la source de l'attaque.
Le travail sur les réponses des IDS est en cours, mais des idées intéressantes commencent à voir le jour. On parle par exemple de dialogue des IDS avec les firewalls, afin de modifier dynamiquement les règles de filtrage suivant les attaques détectées par les NIDS.

4.2.5 Les HIDS complètent l'analyse des NIDS

Nous avons évoqué jusqu'ici principalement le cas des NIDS. Les IDS se cantonnent à la surveillance des systèmes sur lesquels ils sont hébergés. Mais ils sont extrèmement utiles. Par exemple dans le suivi d'une attaque évoqué précédemment, grâce aux sondes NIDS, nous pouvons suivre son parcours. Mais quel est l'impact final sur la machine ? Un NIDS ne peut pas répondre à cela, car il ne gère pas les équipements terminaux. C'est ici que le HIDS se révèle utile. De plus, la remontée d'alerte est locale et vers un manager. Ainsi, la surveillance réseau et des équipements terminaux est centralisée.

4.2.6 Combinaison avec d'autres outils

L'avantage des IDS libres est qu'ils peuvent se combiner avec d'autres outils libres comme Prelude-IDS avec Nessus ou Honeyd, ou encore Snort. Ainsi, la spécificité et l'expertise de chaque outil sont utilisées, les résultats sont combinés, et des relations les uns avec les autres sont étudiées. Nous arrivons alors à une étude très fine des incidents de sécurité, et à une bonne qualité de la sécurité sur le réseau.

4.2.7 Bonne protection des IDS eux-mêmes

Il existe de méthodes de contournement des IDS, mais si le principe est simple, la mise en oeuvre est complexe. Le détail sera donnée plus tard dans ce rapport. Nous avons donc un outil fiable.

4.3 Points faibles de cette méthode de protection

Les IDS ne sont pas là pour remonter des alertes d'attaques involontaires (faute de frappe, erreur de saisie) mais plutot pour détecter des attaques plus élaborées. Toute attaque un minimum préparée, comprend une phase de camouflage ou d'effacement des traces. Sur un système, cela passe par l'effacement des logs ou la modification des attributs des fichiers modifiés. Dans le cas de traces réseaux, cela va passer par l'attaque des sentinelles : les IDS.

Mais d'un autre coté, la simple utilisation des IDS pose quelques problèmes que nous allons maintenant détailler.

4.3.1 Besoin de connaissances en sécurité

La mise en place de sonde sécurité fait appel à de bonnes connaissances en sécurité. L'installation en elle-même des logiciels est à la portée de n'importe quel informaticien. En revanche l'exploitation des remontées d'alertes nécessite des connaissances plus pointues. Les interfaces fournissent beaucoup d'informations, et permettent des tris facilitant beaucoup le travail, mais l'intervention humaine est toujours indispensable. A partir des remontées d'alertes, quelle mesure prendre ? Est-il utile de relever des alertes dont toutes les machines sont protégées (attaques sur MSSQL sur un réseau ayant que du MySQL) ? Comment distinguer un faux-positif d'un véritable incident de sécurité ? Par exemple un nombre important de icmp-redirect peut être le signe d'une attaque de type "homme du milieu", mais aussi d'un routage mal configuré.

La configuration, et l'administration des IDS nécessitent beaucoup de temps, et de connaissances. C'est un outil d'aide, qui n'est en aucun cas complètement automatisé.

4.3.2 Problème de positionnement des sondes

La mise en place est importante. Il faut bien définir là où placer les sondes. Il ne s'agit pas de mettre une sonde partout où l'on veut surveiller. Il faut étudier les champs de vision des sondes suivant leur placement, si on veut recouper ces champs de vision (pour par exemple faire des doublons de surveillance ou faire un suivi d'attaque), quel détail d'analyse (à l'entrée d'un réseau, ou dans chaque domaine de colision). On découpe souvent le réseau global en un LAN, une DMZ, puis Internet. Mais il faut aussi envisager les domaines de collisions, les sous-réseaux, ...

Étant donné que la sonde travaille en mode promiscuité, elle utilise la librairie libpacap ou winpcap qui fait qu'une sonde ne pourra pas être installée sur les firewalls. Et la mise en place sur la sonde même d'un filtrage pour la protéger contre certaines attaques directes aura une efficacité très réduite.

L'utilisation de tunnel est aussi à envisager lors du positionnement des sondes : inutile de placer une sonde où tout le traffic est crypté.

Les connaissances réseaux sont importantes. Il faut aussi faire attention à comment sont remontées les alertes (si on passe par une ligne RNIS, éviter de la monter et la fermer à chaque alerte). Même si la plupart des schémas montrent un manager et N sondes, nous pouvons très bien utiliser M manager et N sondes.

4.3.3 Vulnérabilités des sondes NIDS

De part leur fonctionnement en mode promiscuité, les sondes sont vulnérables. Elles captent tout le traffic, et même si un ping flood est réalisé sur une autre machine, les sondes NIDS le captureront aussi et donc en subiront les conséquences, comme si l'attaque leur était directement envoyée. Les DoS classiques seront donc très nocifs pour les sondes NIDS.

L'analyse par scénario pose aussi des problèmes de contournement avec des attaques par évasion ou par insertion. Le détail sera expliqué plus tard dans ce document.

4.3.4 Un DoS explose les fichiers de logs

Le point fort de certains IDS qui est d'archiver aussi le payload des trames ayant levées une alerte, peut aussi s'avérer un point faible. Un ping flood avec un payload chargé de 64000 octets, ou encore des trames de 1500 octets pour les SYN flood vont faire exploser la taille des fichiers de logs des sondes en quelques minutes. C'est une attaque qui porte le nom *coke* qui consiste à saturer le disque dur (http ://www.securiteinfo.com/attaques/hacking/coke.shtml). La seule façon de parer cette attaque est de prévoir d'importants espaces de stockages, et gérer le stockage des fichiers de logs.

4.3.5 Réponse active non-contrôlée

Certains IDS génèrent une contre-attaque lorsqu'ils détectent des attaques. Le gros problème est de la justification de cette contre-attaque. On pourrait suivre le principe du DDoS pour faire attaquer les sondes. Par exemple, on réalise un ping echo-request-broadcast sur quelques machines, en faisant croire que la source de la demande est une machine du réseau (ou plusieurs si on est très joueur). Toutes les machines vont répondre à cette machine. Ainsi on a un premier niveau de DDoS. Les sondes interviennent alors. Détectant cette attaque comme tentative de surcharge du réseau, elles contre-attaquent la source du ping. On a ainsi un second niveau de DDoS encore plus nocif. En plus d'un engorgement réseau, on a réussi à saturer une machine. Il suffit que cette dernière soit un serveur avec une pile IP faible non protégée, et nous pouvons dire qu'il est planté.

Imaginons maintenant que les contres-attaques des sondes sont vues par l'attaquant (il met son interface en mode promiscuité ou ne camoufle pas le fait qu'il est la source de l'attaque... il préfèrera alors un simple ping flood pour ne pas subir le premier niveau de DDoS). Il va alors pouvoir détecter la présence de sondes IDS, leur nombre, et leur type. Ainsi, il connait avec certitude une partie des mesures de sécurité prises sur tout le parcours de l'attaque du réseau cible ... choses que les responsables sécurité n'aiment pas divulguer. Nous rappellons, comme expliqué précédemment, les problèmes d'explosion des fichiers de logs que ces DDoS entraînent.

Voici un dernier point qui utilise le revers de la médaille des contre-attaques automatiques : si nous mettons 2 sondes sur 2 réseaux séparés A et B, et qu'une attaque se produit sur le réseau B en passant par le réseau A. La conte-attaque de la sonde du réseau B sera vue par la sonde du réseau A. Cette dernière va donc contre-attaquer la source de la première attaque (si on avait été joueur, on aurait falsifié la source en indiquant la sonde du réseau C, réseau qui se trouve derrière B), mais aussi la source de la deuxième attaque qu'est la sonde B. Ceci est une hypothèse, il faudrait tester pour savoir si l'IDS est protégé contre ce type d'attaque. Il y en a surement beaucoup qui ne le sont pas.

4.3.6 Problèmes de IPv4

Comme détaillé plus tard dans ce document, il est facile de contourner certains IDS en utilisant les fragments IP. On exploite ici une faiblesse du protocole IPv4, et des piles IP.

Une autre faiblesse de ce protocole est de ne pas permettre l'authentification. On peut donc faire croire à un destinataire qu'un paquet vient de telle ou telle source alors que cela serait faux. Cela est la base des DoS et DDoS évoqués précédemment.

4.3.7 Problèmes intrinsèques à la plateforme

Beaucoup d'IDS (et plus particulièrement les IDS libres) sont des logiciels reposant sur une système d'exploitation non dédié aux IDS. Ainsi, la faiblesse d'un IDS est liée à la faiblesse de la plateforme. Un même logiciel sera par exemple plus vulnérable sur un PC Win98 que sur un PC OpenBSD, de part la solidité de la pile IP face aux attaques, ou tout simplement de part la stabilité du système. La mise en place d'un IDS requiert donc des compétences dans la sécurisation de la plateforme.

Une saturation de la mémoire, de la carte réseau (éviter des cartes ISA), ou du processeur porte atteinte directement au bon fonctionnement de tout le système et donc du logiciel IDS de la machine. Le problème de ces dysfonctionnements est que si la sonde ne peut plus remplir son rôle, le réseau n'en est pas coupé pour autant (heureusement diront certains). Le responsable sécurité ne peut donc pas voir que la sonde étant tombée, une partie du réseau n'est plus surveillée. Une redondance des surveillances sur certaines zones, devraient momentanément résoudre le problème. Mais contre une attaque bien préparée, c'est inutile.

Malheureusement, nous n'avons pas eu le temps de tester tout cela, afin de connaître les comportements des IDS à leurs limites de fonctionnement, quelles sont ces limites, etc.

4.3.8 Gestion de plusieurs managers très limitée

Il est possible de mettre en place plusieurs managers. Nous avons aussi remarqué la possibilité de gérer plusieurs sondes avec un seul manager. En revanche dans tous les IDS étudiés, il a toujours manqué la gestion de plusieurs managers pour une même sonde. Il serait utile d'envoyer une alerte de type A à un manager X et une alerte de type B à un manager Y. Cela est parfois possible avec quelques bidouillages sur certains IDS, mais comme nous venons de le dire, c'est du bidouillage.

4.3.9 Coût matériel parfois important

La mise en place d'un IDS peut entraîner des coûts matériels importants. Effectivement, sur un grand réseau, le nombre de sondes et de managers peut être important, et donc le nombre de PC ou "de boîtes IDS propriétaires" aussi. Encore une fois, l'utilisation de logiciels libres ou open-sources permet de beaucoup réduire ce problème.

4.4 Méthodes de contournement des IDS

Les systèmes de détection d'intrusion, aussi performants qu'ils puissent être présentent certaines limites(bases de signatures obsolètes, faux positifs, faux négatifs ...).Ces limites peuvent être utilisées pour attaquer ou passer au travers des IDS.

4.4.1 Quelques techniques

Il existe ainsi plusieurs techniques pour échapper à la détection par les IDS notamment dans le cas des IDS par scénario :

L'évasion
C'est faire passer une attaque au travers du système de détection en évitant de correspondre à un scénario répertorié, donc détectable.

L'insertion
C'est l'insertion de trafic qui permet de déjouer l'IDS en lui faisant croire à un trafic légitime. On injecte l'attaque parmi beaucoup d'informations sans incidences. Les signes de l'attaque n'apparaissent donc pas à l'IDS mais quand les données atteignent la cible, seule l'information malintentionnée est acceptée par le système. Cela est très proche du principe des canaux cachés.

D'autres méthodes sont possibles :
- Déni de service
- Flood de signes d'intrusions (faux positifs ou vraies alarmes) pour déclencher beaucoup d'évènements et surcharger les administrateurs. Nous pouvons par exemple utiliser l'outil *snort* pour cela.
- Contournement physique (les trames ne sont pas capturées par les sondes) en jouant sur les domaines de collisions.
- Attaque directe contre l'IDS : comme nous l'avons vu dans ce rapport, les IDS présentent quelques points faibles qui peuvent être exploités (réponse active redirigée, saturation de l'espace de stockage des alertes, faiblesse de la pile utilisée, ...)
- Distribution dans le temps ou dans l'espace

4.4.2 Exemples d'attaques

Pour illustrer ces différentes méthodes, voici quelques exemples d'attaques permettant d'outre-passer les IDS :

Les attaques réseaux
Le but principal est de réduire les possibilités du NIDS à détecter les attaques :

Par les méthodes classiques de scan : Nous pouvons prendre comme exemple le scan furtif SYN implémenté par NMAP permettant de ne pas être détecté par les NIDS. Le but du scan SYN est de ne pas ouvrir une connexion complètement. A la réception d'un SYN/ACK qui signifie que le port est ouvert, il envoie un RST pour interrompre la connexion. Aucune connexion n'est donc faite tout en sachant quels ports sont ouverts.

Par le flood : Il consiste à surcharger le NIDS pour qu'il ne puisse pas fonctionner correctement, et qu'il ne détecte pas l'attaque principale.

Par la noyade sous les faux-positifs : Le principe est de provoquer de nombreuses remontées d'alertes. En parallèle, une attaque réelle contre le réseau est lancée, et l'administrateur occupé à analyser les alertes, ne s'en rendra pas compte sur le moment. Nous pouvons utiliser l'option *decoy* de nmap pour générer des scans nmap multiples, ou encore utiliser l'outil *snot* qui génère de nombreuses attaques différentes.

Par fragmentation : Le principe est de fragmenter les paquets IP pour empecher les NIDS de détecter les attaques sachant que les paquets seront réassemblés au niveau du destinataire. Il est possible aussi d'envoyer des paquets IP mal fragmentés qui vont utiliser la faiblesse de certaines pile IP (peut-être aussi celle de la sonde IDS qui capte tout le trafic) pour perturber le système.

Par des scans lents : Les NIDS ne détectent souvent pas ce type de scan (un scan toutes les heures par exemple). Sachant qu'un NIDS maintient un état de l'information (TCP, IP fragments, TCP scan, ...) pendant une période de temps bien définie (capacité mémoire, configuration), ils ne détectent rien si deux scans consécutifs sont trop espacés.

Les techniques de RFP

Plusieurs techniques anti-IDS ont été développées par Rain Forest Puppy ou RFP au niveau du protocole HTTP implémenté dans son scanner cgi Whsiker. Le principe est de lancer les attaques sous une forme différente de celles référencées dans la base de signatures des IDS afin de ne pas reconnaître les requêtes HTTP. De plus, il rend les attaques complexes pour qu'elles ne puissent pas être détectées. Nous allons lister ci-après quelques techniques :

l'encodage : Il permet de coder les caractères sous forme hexadécimale.L'URL sera comprise par le protocole HTTP.

les doubles slashes : Par exemple, une requete de type '//cgi-bin//script' ne sera pas détectée car les IDS vérifient les requêtes de la forme '/cgi-bin/script'.

les self-reference directories : Il consiste à remplacer tous les '/' par des '/./'. La requête n'est pas détectée.

La simulation de fin de requête : L'IDS analyse la première partie de l'URL et s'arrête au premier HTTP/1.0 \r\n. Le reste de la requete qui représente l'attaque passe sans etre analysée. Par exemple : HEAD /HTTP/1.0\r\nHeader : /../../cgi-bin/some.cgi HTTP/1.0\r\n\r\n

Le formatage : Il consiste à remplacer les espaces par des tabulations.

Case sensitive : Il consiste à remplacer les minuscules par des majuscules. La requête reste valide.

URL coupée : Il consiste à couper la requete HTTP en plusieurs paquets TCP. Par exemple, l'URL "GET /cgi/bin/phf" devient "GET","/cgi/b","in/ph","f HTTP/1.0".

Le caractère NULL : En analysant la chaine de caractères HEAD%00 /cgi-bin/some.cgi HTTP/1.0, l'IDS s'arrête dès qu'il atteind le caractère NULL, la suite de l'URL n'étant pas analysée.

Buffer-overflow

Les IDS sont capables de détecter des attaques de type buffer overflow.Pour cela, il analyse le trafic à la recherche de chaines de caractères telles que "/bin/sh", "0x90"(NOP)...

4.5 Une alternative aux IDS

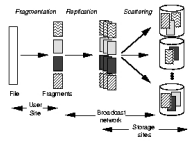
4.5.1 La tolérance d'intrusion

Il est difficile de protéger la disponibilité des données et des services dans la plupart des attaques vues précedemment. En revanche, nous pouvons faire quelque chose dans la protection de l'intégrité et de la confidentialité des données : c'est la tolérance d'intrusion.
Elle est définie comme suit : "Un système distribué à tolérance d'intrusion est un sytème dont le but est de ne pas mettre en danger la confidentialité, l'intégrité en cas d'intrusion dans une partie du système."

Ce concept de tolérance d'intrusion est implémenté via la technique "Fragmentation-Redundancy-Scattering" :

Elle consiste à éclater en plusieurs fragments l'information (données, programmes, droits d'accès) avant de l'enregistrer sur des sites géographiquement différents. Grâce à cette méthode de fragmentation-scattering, un certain nombre d'intrusions est toléré tout en respectant une intégrité et une confidentialité de l'information.

A cette tolérance d’intrusions, on associe une tolérance de destructions d’information en dupliquant les fragments. Plusieurs copies de ces fragments sont archivées sur plusieurs sites. Ceci assure une disponibilité de l’information.



4.6 Conclusion

Les IDS continuent d’évoluer pour répondre aux exigences technologiques du moment mais offrent d’ores et déjà un éventail de fonctionnalités capables de satisfaire les besoins de tous les types d’utilisateurs. Cependant comme tous les outils techniques ils ont des limites que seule une analyse humaine peut compenser. Les détecteurs d’intrusions deviennent chaque jour meilleurs grâce à l’expérience acquise avec le temps. Mais ils deviennent aussi de plus en plus sensibles aux erreurs de configuration et de paramétrage. Par conséquent, il est plus que fondamental de former correctement les personnes chargées de la mise en oeuvre et de l’exploitation des IDS.

5 Prelude-IDS et ses compères

.

5.1 Historique

Le projet de Prelude a commencé en 1998 et avait pour but de créer un outil modulaire de détection d'intrusion réseau composé d'une sonde et d'un Report Server. Lors du Libre Software Meeting 2001, les équipes de Prelude et du projet Trithème (projet indépendant lancé en février 2000) ont décidé de joindre leurs efforts dans le but d'évoluer progressivement vers le développement d'un IDS hybride basé sur la prise en compte de la quasi-totalité des évènements sécurité au niveau réseau (Network-based IDS) et local (Host-based IDS) grâce à des sondes dédiées.

5.2 Caractéristiques

5.2.1 Description

Prelude-IDS (Intrusion Detection System) est un système de détection d'intrusions et d'anomalies distribué sous licence GPL.

Un tel système vient compléter la panoplie des équipements et logiciels de sécurité (serveurs proxy, routeurs filtrants, firewalls...) et offre à l'analyste un outil de contrôle des activités suspectes ou illicites (interne comme externe).

La détection d'intrusion est réalisée par l'analyse du trafic réseau et l'utilisation de signatures d'évènements hostiles ou par l'analyse en continue de fichiers de journalisation.

L'architecture de Prelude est modulaire (on peut intégrer ou développer de nouvelles fonctionnalités grâce à des plugins), distribuée (Prelude est une suite de composants autonomes et interactifs, ie les sondes et les managers) et sécurisée (utilisation du support SSL pour l'authentification et le chiffrement des communications).
Les sondes (réseaux comme locales) n'effectuent que les opérations de surveillance et de génération d'alertes alors que les managers prennent en charge la gestion des sondes et la journalisation des alertes.

5.2.2 Architecture

- Les capteurs sont des entités de détection capables de remonter des alertes à un manager Prelude.
- Le manager (il peut y en avoir plusieurs) accepte les connexions en provenance des différents capteurs et collecte leurs alertes. Il assure les fonctions suivantes :
  Le logging qui permet de transformer une alerte au format Prelude en un format lisible par l'analyste.
  La contre-mesure qui permet à l'utilisateur de définir une réaction à une attaque.
- Les agents de contre-mesure sont placés sur les machines devant opérer la réaction à une attaque (fonction en cours de développement).
- Le frontend est une interface d'administration web permettant d'aider les administrateurs sécurité à analyser plus facilement les alertes et les statistiques, sachant que cette tâche ne peut être complètement automatisée à l'heure actuelle.
- La communication entre les différents programmes se fait au format IDMEF (Intrusion Detection Message Exchange Format). Ce format fondé sur XML est suffisamment générique pour permettre aux composants hybrides de Prelude d'émettre des alertes décrivant des événements de tous types : attaques réseau, buffer-overflow local, ...

5.2.3 Composants

Libprelude (la librairie Prelude)
La librairie `libprelude` constitue la brique de base de tout composant Prelude à l'exception du frontend. Cette librairie fournit aux composants Prelude les fonctionnalités suivantes :
– Gestion de la connexion entre composants (sondes et managers) notamment le mécanisme de reprise après interruption et de rétablissement automatique de la connexion ;
– Gestion du mode de communication entre composants, notamment la prise en charge du chiffrement éventuel et de l'authentification ;
– Interface permettant l'intégration de plugins.
Cette librairie doit être installée préalablement à l'installation de tout autre composant (à l'exception toujours du frontend).
Nous utilisons la version 0.8.4.

Prelude-Nids (la sonde réseau)
Cette sonde prend en charge l'analyse en temps réel du traffic réseau.
Elle est construite au-dessus de la librairie `libprelude` et fournit :
– Un moteur de gestion de signatures générique, actuellement compatible avec les signatures Snort, mais pouvant être étendu par l'ajout de nouveau "parser" de règles ;
– Des modules spécialisés par protocole : par exemple, un plugin est dédié aux protocoles RPC et permet l'analyse fine de ce type de connexions ;
– Des modules spécialisés dans la détection non basée sur des signatures : détection des activités de balayage (scan), ...

Les sondes réseaux peuvent aussi prendre en charge la défragmentation IP et le réassemblage des flux TCP, de façon à rendre une sonde réseau Prelude moins vulnérable aux attaques de type Snot.
Nous utilisons la version 0.8.1.

Prelude-LML (la sonde locale)
Cette sonde prend en charge la remontée d'alertes détectées localemenent sur une machine. Cette détection est basée sur l'application à des objets (fichiers de journalisation et/ou application) de règles construites autour d'expressions régulières compatibles Perl (PCRE).
Pour la surveillance des systèmes Unix, une sonde prelude-LML peut utiliser le service syslog et ainsi assurer la remontée d'alertes. L'intégration des systèmes Microsoft peut également se faire à l'aide de l'utilitaire ntsyslog.
Un message est généré par la sonde prelude-LML dès qu'une ligne de log correspond à une expression régulière.
Nous utilisons la version 0.8.2.

Prelude-Manager (le contrôleur)
Prelude-manager centralise les messages des sondes réseaux et locales, et les traduit en alertes.
Il est responsable de la centralisation et de la journalisation à travers deux fonctions :
– Celle de relais : un contrôleur relais va assurer le routage vers un contrôleur maître d'alertes provenant des sondes qui lui sont rattachées.
– Celle de maître : un tel contrôleur va assurer la réception des messages et des alertes provenant des sondes ainsi que leur journalisation dans un format lisible par l'analyste : en mode texte (dans les fichiers) ou SQL dans le cas de l'utilisation d'un SGBD (MySQL ou PostgreSQL).

Il est possible d'étendre les capacités d'un contrôleur à l'aide de plugins, en autorisant, par exemple, le traitement de messages en provenance de composants autres que Prelude, un contrôleur Prelude pouvant ainsi centraliser la remontée d'alarmes en provenance de sondes Snort.
Nous utilisons la version 0.8.6.

Prelude-Frontend (l'interface web)
C'est l'interface de visualisation des alertes. Il est actuellement proposé deux interfaces, l'une développée en PHP et l'autre en Perl.

– Prelude-PHP-Frontend C'est l'interface proposée sur le site "www.prelude-ids.org". Elle est composée de scripts PHP et est destinée à être installée sur un serveur web indépendamment des autres composants Prelude. Cela signifie que l'installation préalable de la librairie `libprelude` est inutile, mais que par contre celle d'un serveur web supportant PHP4 l'est.
  Nous utilisons la version 0.8.1.
– Prelude-Perl-Frontend Cette interface est issue d'un projet intitulé "Le Routier" (`www.leroutier.net/Projects/PreludeIDS`). Elle nécessite bien évidemment l'installation d'un serveur web supportant Perl.

5.3 Prelude-IDS et Snort

Nous allons étudier les points communs et les différences entre ces deux outils en commencant par évoquer les projets qui les ont fait naître. Puis nous partirons du niveau le plus bas qu'est le moteur d'analyse, pour finir à l'interface de remonté des alertes. Enfin, nous dirons quelques mots sur la facilité de configuration, et la possibilité d'intégration d'outils externes au projet.

5.3.1 Projet

Dans le cadre du projet, ces deux outils sont très proches. Ce sont tous les deux des outils libres. Les projets dont ils sont les résultats sont très actif, aussi bien dans le développement que dans la mise à jour des attaques.

Quelques avantages de Snort sur Prelude-IDS sont sa popularité et sa disponibilité sur de nombreuses plateformes. Prelude-IDS se restreint sur les plateformes POSIX alors que nous pouvons retrouver Snort aussi sur Windows. A cela s'ajoute l'importance de la base de données des signatures d'attaque de Snort, pour expliquer sa plus grande popularité. Mais Prelude-IDS est de plus en plus connu dans le monde des professionnels de la sécurité.

Une différence importante cependant, au désavantage de Snort : Snort est un NIDS pur alors que Prelude-IDS intègre des fonctionnalités NIDS et HIDS. Ce dernier est un IDS hybride. Dans la suite, nous ne comparerons que ce qui est comparable, c'est-à-dire la partie NIDS de Prelude-IDS et Snort.

5.3.2 Moteur d'analyse et banque de signatures

Les deux font des analyses par recherche de similitudes avec un scénario préalablement définis. Le mode de fonctionnement est alors similaire. Autant pour Prelude-IDS que pour Snort, les solutions sont stables.

Prelude-IDS a un soucis de suivi des standards (pour pouvoir utiliser les signatures de d'autres moteurs, échange des messages en XML, IDMEF).

Notons que même en intégrant les alertes de Snort, Prelude-IDS relève une quantité plus importante d'alertes. En regroupant les alertes de même type, nous nous retrouvons alors avec aproximativement le même nombre d'alertes (Prelude ayant ses propres règles, il est normal qu'il en trouve un peu plus que Snort). Prelude-NIDS archive aussi les payloads. Mais malheureusement, la fusion des payloads (ou des alertes) n'est pas plus présente dans Prelude-IDS que dans Snort.

Prelude-IDS a l'avantage d'être très modulaire de par son architecture client-serveur. Un manager peut gérer plusieurs sondes, et une sonde peut envoyer ses alertes à plusieurs managers. Le filtrage **au niveau de la sonde** pour savoir à quel manager est envoyée telle alerte ou telle autre, n'est pas encore

opérationnel. Mais nous avons envoyé un email sur la mailing list du projet, qui a semblé susciter de l'intérêt pour cette fonctionnalité. Peut-être qu'elle sera donc prochainement implémentée.

5.3.3 La remontée d'alertes

En utilisant ces outils, nous nous rendons compte du fait que Prelude-IDS remonte un nombre plus important d'alertes que Snort. Là où Snort remonte 30 alertes de 2 types différents, Prelude-IDS remonte parfois 2000 alertes de 3 types différents. Cela est dû au fait que Prelude relève pratiquement chaque trame suspecte, alors que Snort ne va pas relever chaque trame d'une connexion suspecte. De plus, si on combine les signatures d'attaque de Snort avec celles de Prelude, on se rend compte que Prelude à la capacité de détection d'un plus grand nombre d'attaque. Cela est logique étant donné qu'il s'appuit sur les mêmes signatures que Snort en plus des siennes.

5.3.4 Infos disponibles dans les alertes

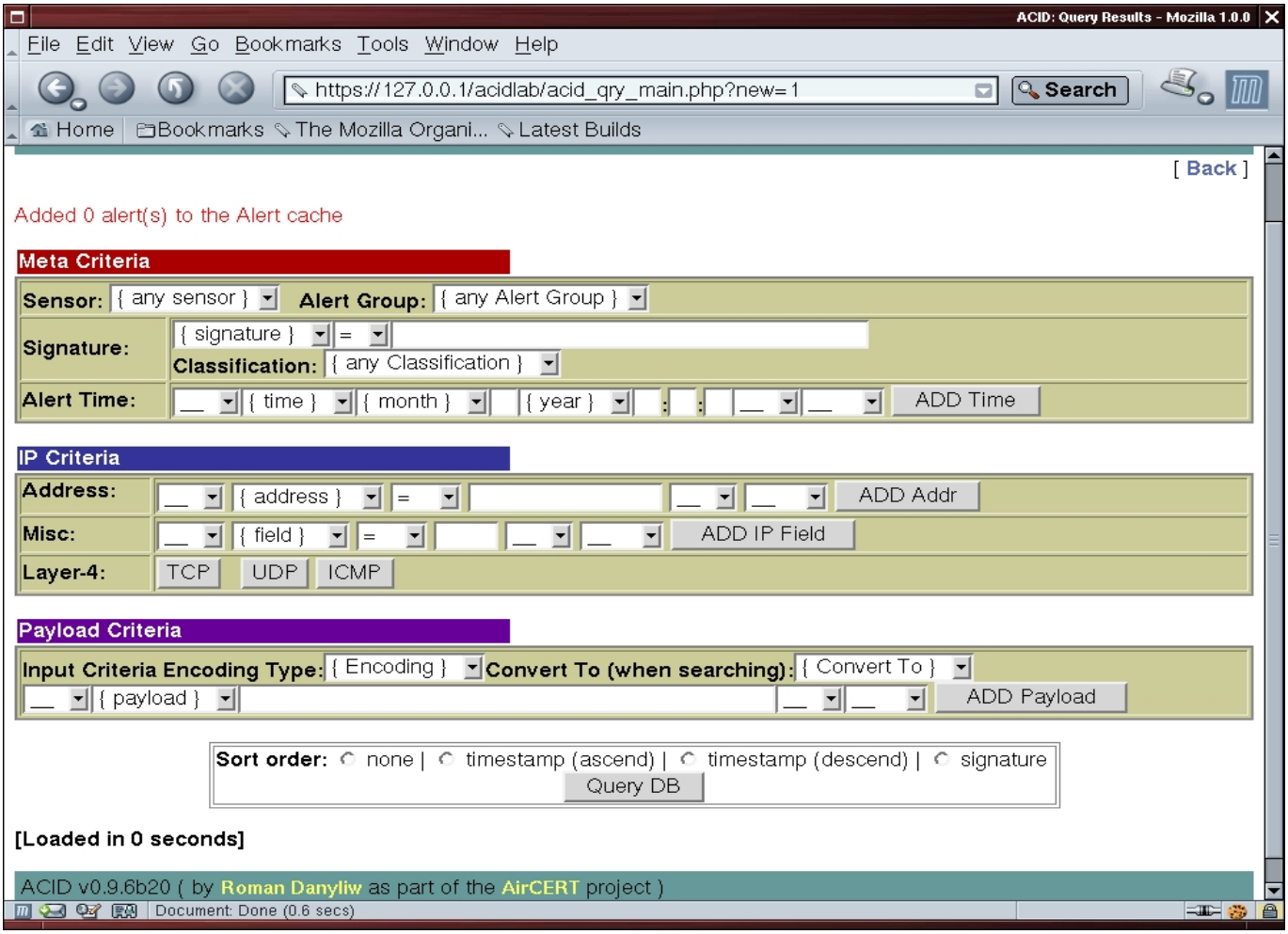
Dans les deux cas nous avons des remontées d'alertes bien détaillées, qui permettent de connaître la source et la destination d'une attaque, le type d'attaque, un premier classement de sévérité de l'alerte, ou encore un lien vers un bulettin d'alerte officiel.
Dans les deux outils, il est intéressant de constater la présence de niveaux d'alertes permettant d'en déduire du premier coup d'œil la criticité.

Il est à noter que Prelude-IDS archive aussi les charges utiles des trames suspectes. Cela permettra ensuite dans certains cas à l'administrateur d'analyser le contenu afin d'écarter plus facilement les faux-positifs.

5.3.5 Les frontends

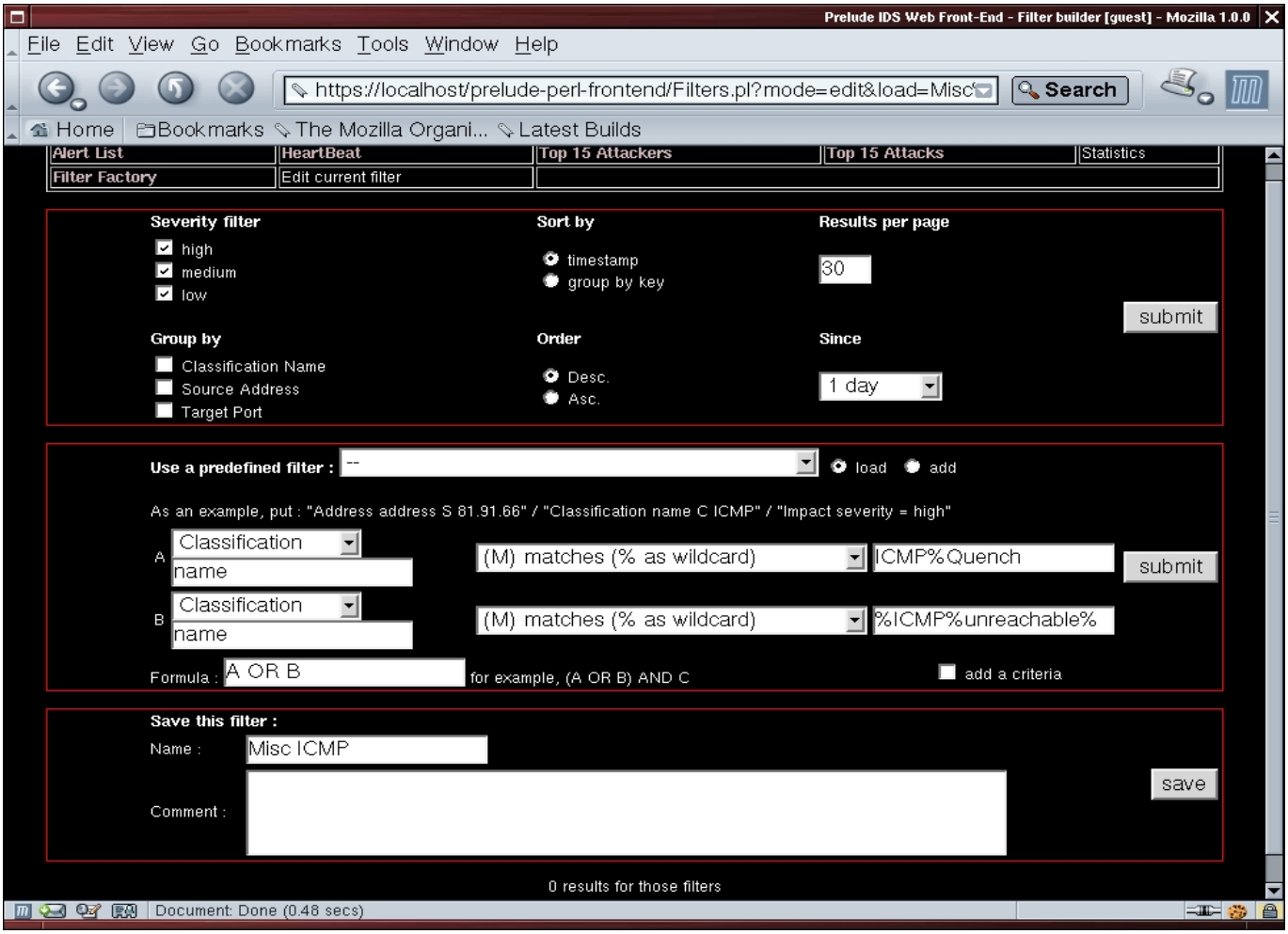
Les plus grandes différences entres les 2 outils se trouvent au niveau des frontends.

– Nombre de frontends disponibles : Sous Prelude, il n'y a pas vraiment de frontends officiel. Lors du début du TER l'ancien frontend venait d'être laissé à l'abandon pour un nouveau frontend php. Puis quelques semaines après, un nouveau frontend en perl arrivait à maturité pour le remplacer. Il y en a encore d'autres, mais c'est le frontend perl qui est en ce moment le plus abouti et le plus riche. C'est donc lui que nous allons comparer avec le frontend dédié à Snort : ACID.
  Snort compte lui aussi plusieurs interfaces, et sa popularité fait qu'il est maintenant intégré dans des outils comme webmin. Cependant, la référence reste ACID.
– Classement et regroupement des alertes : Dans les deux cas, le regroupement des alertes suivant des critères communs basiques (même adresse IP source, même type d'attaque, même niveau d'alerte, ...) est équivalent. Pour des filtrages plus détaillé, la logique est différente dans les deux outils. ACID permet de trier les alertes en choisissant plusieurs critères.


ACID: Query Results - Mozilla 1.0.0
File Edit View Go Bookmarks Tools Window Help
https://127.0.0.1/acidlab/acid_qry_main.php?new=1
Search
Home Bookmarks The Mozilla Organi... Latest Builds
[Back]
Added 0 alert(s) to the Alert cache
Meta Criteria
Sensor: { any sensor } Alert Group: { any Alert Group }
Signature: { signature } =
Classification: { any Classification }
Alert Time: { time } { month } { year } ADD Time
IP Criteria
Address: { address } = ADD Addr
Misc: { field } = ADD IP Field
Layer-4: TCP UDP ICMP
Payload Criteria
Input Criteria Encoding Type: { Encoding } Convert To (when searching): { Convert To }
{ payload } ADD Payload
Sort order: none | timestamp (ascend) | timestamp (descend) | signature
Query DB
[Loaded in 0 seconds]
ACID v0.9.6b20 (by Roman Danyliw as part of the AirCERT project)
Document: Done (0.6 secs)

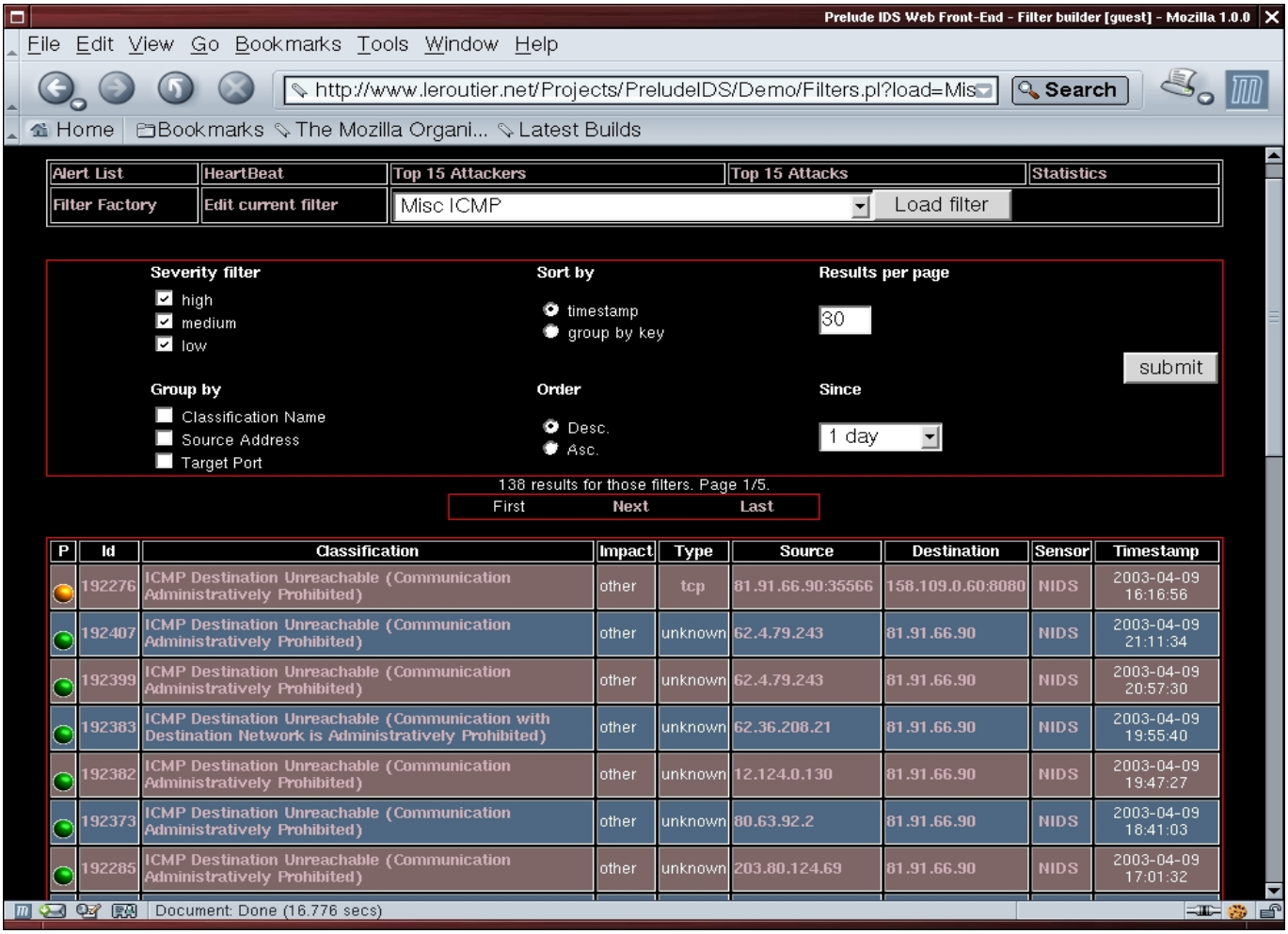
Filtres d'ACID

En revanche, le frontend-perl de Prelude-IDS propose une rubrique *Filter Factory* qui permet de créer des filtres, de les modifier, ou de les supprimer.



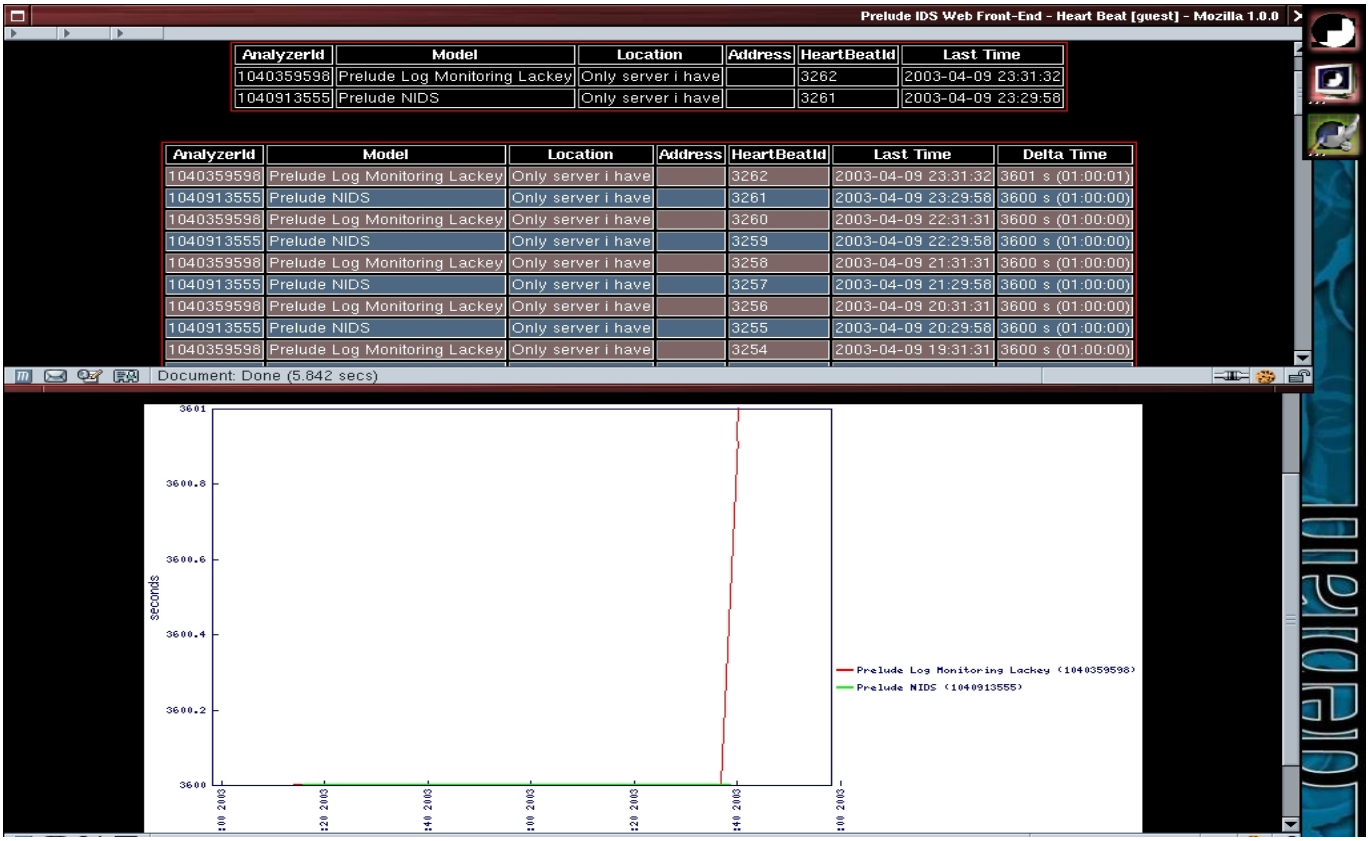
La rubrique Filter Factory

Ces filtres sont ensuite disponibles dans la page principale (*Alert List*). Ce qui est intéressant avec le *Filter Factory* c'est que les filtres ainsi créés sont sauvegardés.

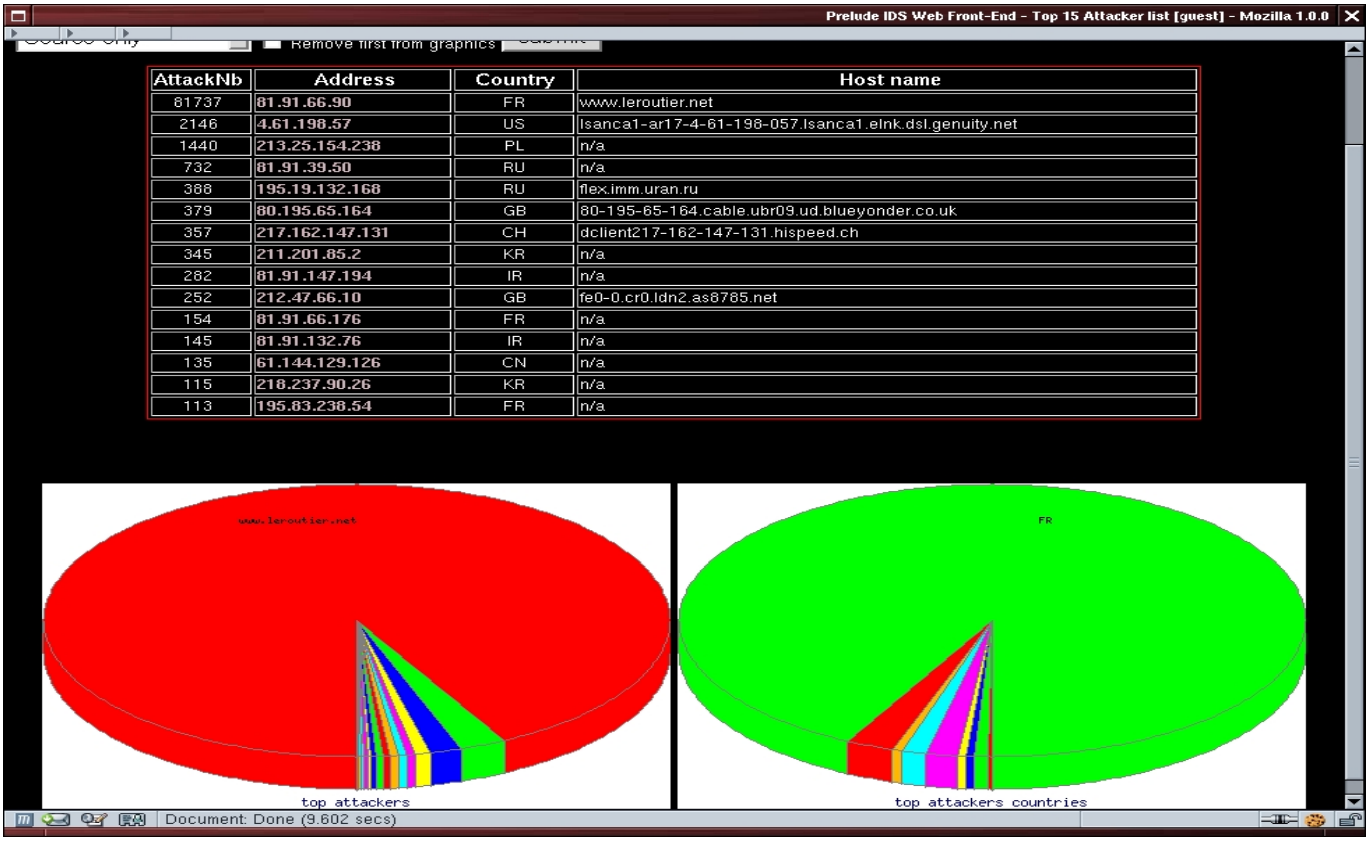


Rubrique Alert List avec le filtrage défini précédemment

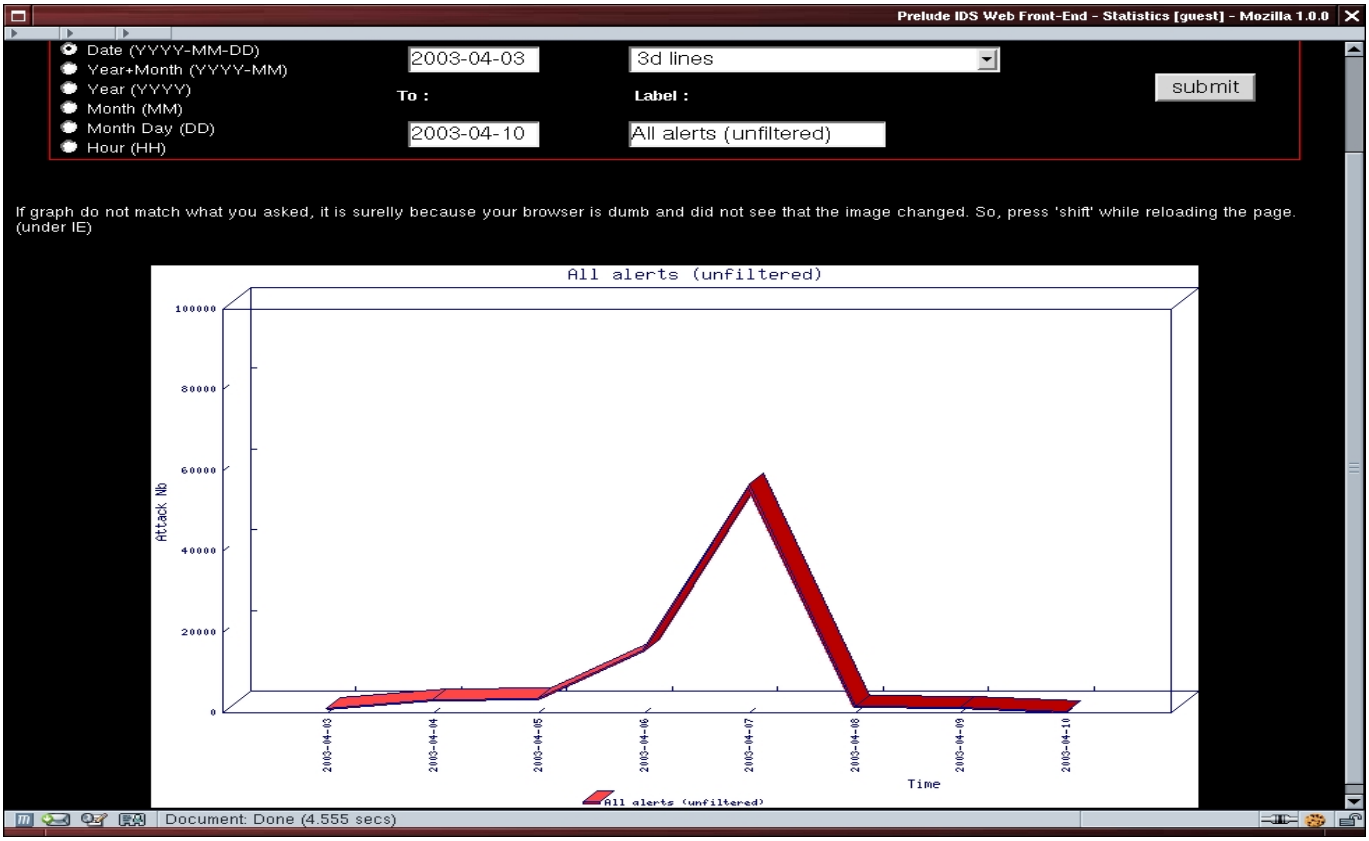
– Graphiques : Le frontend-perl de Prelude-IDS permet la visualisation de nombreuses informations sous forme de graphiques. Cela va du top 15 des attaquants, à la surveillance des sondes (*hearthbeat*), en passant par le top 15 des attaques, ou encore des statistiques des attaques. De plus il est possible dans certains cas de choisir d'afficher sous forme de camembert, d'histogramme, ligne fine, tableau, ... C'est un vrai plaisir de manipuler cette interface. ACID est sur ce point moins convivial, et même si l'affichage basique des alertes (sous forme d'un tableau) est équivalent, le frontend-perl est beaucoup plus riche pour le reste.


Prelude IDS Web Front-End - Heart Beat [guest] - Mozilla 1.0.0
AnalyzerId | Model | Location | Address | HeartBeatId | Last Time
1040359598 | Prelude Log Monitoring Lackey | Only server i have | | 3262 | 2003-04-09 23:31:32
1040913555 | Prelude NIDS | Only server i have | | 3261 | 2003-04-09 23:29:58
AnalyzerId | Model | Location | Address | HeartBeatId | Last Time | Delta Time
1040359598 | Prelude Log Monitoring Lackey | Only server i have | | 3262 | 2003-04-09 23:31:32 | 3601 s (01:00:01)
1040913555 | Prelude NIDS | Only server i have | | 3261 | 2003-04-09 23:29:58 | 3600 s (01:00:00)
1040359598 | Prelude Log Monitoring Lackey | Only server i have | | 3260 | 2003-04-09 22:31:31 | 3600 s (01:00:00)
1040913555 | Prelude NIDS | Only server i have | | 3259 | 2003-04-09 22:29:58 | 3600 s (01:00:00)
1040359598 | Prelude Log Monitoring Lackey | Only server i have | | 3258 | 2003-04-09 21:31:31 | 3600 s (01:00:00)
1040913555 | Prelude NIDS | Only server i have | | 3257 | 2003-04-09 21:29:58 | 3600 s (01:00:00)
1040359598 | Prelude Log Monitoring Lackey | Only server i have | | 3256 | 2003-04-09 20:31:31 | 3600 s (01:00:00)
1040913555 | Prelude NIDS | Only server i have | | 3255 | 2003-04-09 20:29:58 | 3600 s (01:00:00)
1040359598 | Prelude Log Monitoring Lackey | Only server i have | | 3254 | 2003-04-09 19:31:31 | 3600 s (01:00:00)
Document: Done (5.842 secs)
seconds
Prelude Log Monitoring Lackey (1040359598)
Prelude NIDS (1040913555)

Surveillance de l'état des sondes

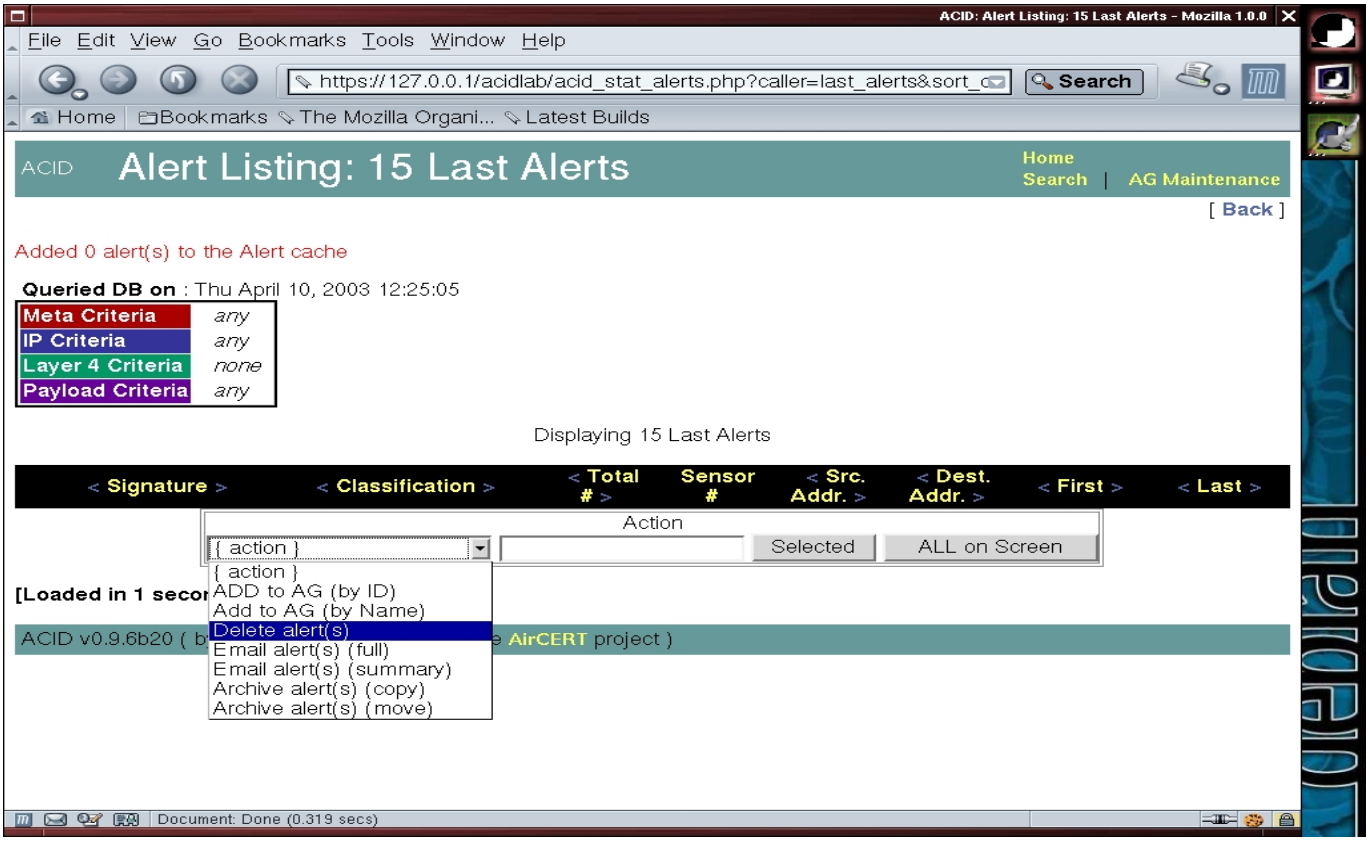

Prelude IDS Web Front-End - Top 15 Attacker list [guest] - Mozilla 1.0.0
Remove first from graphics
AttackNb | Address | Country | Host name
81737 | 81.91.66.90 | FR | www.leroutier.net
2146 | 4.61.198.57 | US | lsanca1-ar17-4-61-198-057.lsanca1.elnk.dsl.genuity.net
1440 | 213.25.154.238 | PL | n/a
732 | 81.91.39.50 | RU | n/a
388 | 195.19.132.168 | RU | flex.imm.uran.ru
379 | 80.195.65.164 | GB | 80-195-65-164.cable.ubr09.ud.blueyonder.co.uk
357 | 217.162.147.131 | CH | dclient217-162-147-131.hispeed.ch
345 | 211.201.85.2 | KR | n/a
282 | 81.91.147.194 | IR | n/a
252 | 212.47.66.10 | GB | fe0-0.cr0.ldn2.as8785.net
154 | 81.91.66.176 | FR | n/a
145 | 81.91.132.76 | IR | n/a
135 | 61.144.129.126 | CN | n/a
115 | 218.237.90.26 | KR | n/a
113 | 195.83.238.54 | FR | n/a
www.leroutier.net
FR
top attackers
top attackers countries
Document: Done (9.602 secs)

Top 15 attaquants



Historique du nombre d'attaques

– Maintenance de la base : Les alertes peuvent être stockées dans un fichier ainsi que dans une base de données. Il est intéressant de pouvoir administrer son contenu au travers de l'interface. ACID le permet, avec possibilité, par exemple, de supprimer un ou plusieurs enregistrements d'un coup. Le frontend-perl en est incapable, et le responsable sécurité devra directement agir sur la base de données avec des requêtes SQL pour la maintenance. Le frontend-perl se cantone à l'affichage, avec classements, affichage sous forme de graphiques, mais au final cela ne sera que de l'affichage. Le frontend-perl étant récent et en plein développement, on peut espérer que ces fonctionnalités seront bientôt implémentées ... même si cela n'apparaît pas pour le moment dans le TODO.
  La maintenance de la base au travers du frontend aurait été particulièrement utile dans Prelude-IDS, qui a tendance à enregistrer un nombre très important d'alertes se ressemblant beaucoup.



Maintenance de la base de données avec ACID

5.3.6 Intégration d'outils externes

La grande force de Prelude-IDS est de pouvoir intégrer les fonctionnalités d'autres outils de sécurité de référence. On peut, par exemple, utiliser Honeyd comme une sonde, envoyer les résultats vers le manager qui les intègrera ensuite dans la base de donées.

La banque de scénario de Snort peut aussi être récupérée par Prelude-NIDS et ajoutée aux règles de Prelude-IDS. Ainsi, le nombre important de signatures reconnues par Snort (et qui participe à sa popularité) bénéficie à Prelude-IDS.

La corrélation des alertes de Prelude-IDS et des failles détectées par Nessus est très intéressante.

Il y a aussi libsafe, ou encore Systrace pour la sécurité du logiciel ...

5.3.7 Facilité de configuration et bonne documentation

Les deux outils sont facilement configurables. Cela ne passe pas toujours par une interface graphique (surtout pour Prelude), mais les fichiers de configuration sont simples. Cela requiert biensûr un minimum de compétences, mais on serait tenté de dire que si on n'arrive pas à configurer l'outil sans clickodrome, ce n'est pas la peine d'aller plus loin car on ne saura pas bien l'utiliser ainisi que les informations qu'il aura remonté.

La documentation sur Snort est très importante et complète. La documentation de Prelude est beaucoup moins importante, parcemée, et contient parfois des erreurs. Mais avec ce rapport, cela devrait changer pour les francophones :-)

5.4 Prelude-IDS et LogCheck

Nous allons ici comparer les logiciels Logcheck et Prelude-lml, tous deux HIDS.

5.4.1 Présentation de LogCheck

Logcheck est un composant du projet Abacus de développement d'outils de sécurité. Ce programme est destiné à faciliter le traitement des journaux système UNIX générés par les démons systèmes, les paquets TCP Wrapper, ...

Logcheck facilite la mise en évidence automatique des problèmes et des violations de sécurité dans les journaux, et envoie les résultats par courrier électronique.

5.4.2 Fonctionnement de LogCheck

Le principe de fonctionnement est le suivant : un script est lancé par cron, il scrute le contenu des logs et retourne les lignes suspectes à une adresse email définie.

Le programme principal utilise une série de fichiers contenant des mots-clés qui sont susceptibles de représenter une tentative d'intrusion comme *refused connect*. D'autres fichiers contiennent les mots à ne pas considérer comme une attaque. Ce programme contrôle les journaux systèmes avec une simple commande `grep`.

5.4.3 Installation de LogCheck

LogCheck est un package Debian facile à installer (logcheck 1.1.9.8) :

```
# apt-get install logcheck
```

5.4.4 Comparaison entre Prelude-IDS et Logcheck

Prelude-lml est le composant de Prelude-IDS que l'on peut comparer avec Logcheck.
Prelude-lml peut être utilisé comme un serveur Syslog (à l'écoute de messages Syslog transitant sur le réseau) ou comme un analyseur de logs. De son coté, Logcheck est un simple analyseur de logs.
La méthode utilisée par Prelude-lml et Logcheck est identique : une alerte est générée quand une expression régulière est présente dans un log. La différence est que Logcheck contient une liste de mots clef qui doivent être ignorés et qui ne lèvent pas d'attaque.

Pour les remontées d'alertes, Logcheck utilise les emails alors que Prelude-lml remonte ses alertes à la fois dans des fichiers de logs spécifiques et dans une base de données (sur un manager Prelude). Cette base de données peut alors être consultée via l'interface Prelude-frontend.

Notons que l'on ne peut pas intégrer Logcheck dans Prelude-IDS. Pourtant, il est possible d'utiliser Logcheck sur les fichiers de logs générés par Prelude.

5.5 Prelude-IDS et Nessus

5.5.1 Nessus, généralités

Présentation
Nessus est un scanner de sécurité multi-plateformes reconnu : il réalise un audit de sécurité sur les différents composants du réseau. En s'appuyant sur une base de données de failles de sécurité, il indique à l'administrateur réseau les faiblesses et les failles existantes sur son réseau. Il est basé sur une architecture client-serveur.

Nessus est un outil capable de répondre à ces questions :
– Quelles machines sont présentes sur mon réseau (adresse IP, nom) ?

– Quel système d'exploitation est utilisé sur une machine donnée ?
– Quels services (démons) fonctionnent sur la machine ?
– Mon réseau est-il sûr aujourd'hui ?

Fonctionnement de Nessus
Nessus permet un audit de sécurité en s'appuyant sur deux éléments : un client et un serveur. Le serveur, `nessusd`, est chargé de tester le système indiqué en essayant toutes les attaques que sa base contient, pendant que le client, `nessus` (qui n'est qu'une interface graphique), fait un rapport sur les différents résultats obtenus.
Le serveur possède une base de données d'environ 300 attaques existantes à l'installation et c'est à l'administrateur de la remettre à jour régulièrement. Les attaques mises en place par `nessusd` sont codées comme des modules externes (ou plugins) écrits en différents languages.
Les communications entre le client et le serveur sont cryptées.

Caractéristiques de Nessus
Nessus permet :
– une reconnaissance intelligente des services : Nessus ne se base pas sur les ports pour reconnaître les services. Ainsi un serveur HTTP fonctionnant sur un port 1234 sera détecté.
– un support du français et de l'anglais.
– une détection sur plusieurs machines de manière concurrente afin d'accélérer l'analyse.

Visualisation des résultats de Nessus :
Les analyses faites par Nessus peuvent être visualisées sous forme de fichiers HTML, LaTeX, et XML notamment.

5.5.2 Comparaison entre Nessus et Prelude-IDS

Tout d'abord, Prelude permet de détecter les attaques lancées sur un réseau alors que Nessus est chargé de relever les failles que sont suceptibles d'exploiter ces attaques. Nessus travaille de façon statique en observant des failles à un moment t (failles de sécurité liées au mauvais fonctionnement ou au mauvais codage d'une application) pendant que Prelude travaille de façon dynamique en observant les flux d'informations transitant sur le réseau. De ce fait, un administrateur utilise plutôt Nessus pour pouvoir prévenir les attaques en appliquant des patchs correctifs tandis qu'il utilise Prelude pour avoir une réaction la plus efficace possible après une attaque.

L'utilisation première de Prelude est de fonctionner en continu sur un réseau alors que Nessus est normalement utilisé par l'administrateur pour une vérification ponctuelle (même si l'utilisation de Nessus évolue en ce moment vers un outil de surveillance "permanent").

Au niveau de l'architecture, Prelude requiert une architecture client-serveur avec l'installation de sondes (réseaux ou hôtes) et d'un manager. Au contraire, Nessus fonctionne sur un seul poste. On peut relever un point commun sur l'interface graphique qui, dans les deux cas, est une partie indépendante de l'application.

Notons que Prelude repose sur une plateforme de type Unix ou BSD. Mais des logiciels comme Ntsyslog nous permettent de remonter des informations sur des machines Windows. En revanche, Nessus fonctionne aussi bien sous Unix que Windows.

L'installation et l'étude des corrélations entre les résultats de Nessus et Prelude-IDS sont détaillés en annexe. Dans notre étude, nous avons eu le résultat suivant :

```
manager:/home/prelude# ability-finder.pl -B -c vuln.conf
*** Trying to connect to the database
 type:mysql      database:prelude
 hostname:localhost      account:prelude

*** Parsing the database to get the alerts id

*** Trying to search for correlation
*** Operation of vulnerability correlation search finnished

 Number of alerts treated in the database : 1073
 Number of correlations of vulnerability  : 0
manager:/home/prelude#
```

Ici on peut voir qu'il n'y a eu aucune attaque exploitant une vulnérabilité préalablement détectée par Nessus.
Il existe aussi un script très simpliste pour faire un tri sur les corrélations durant un laps de temps donné. Par exemple, pour afficher les corrélations pour les dernières 24 heures (ou les 86400 dernières secondes) :

```
manager:/home/prelude/# ./prelude-correlation-agent.pl -B -p 86400
```

Le détail est une nouvelle fois en annexe.

5.5.3 Conclusion

Pour le moment, les règles d'analyse de Nessus ne sont pas intégrées dans Prelude. Cela est en prévision comme indiqué dans la présentation Prelude, réalisée pour la FOSDEM
(http ://www.prelude-ids.org/download/misc/fosdem/2003/slides/html/).

Les scripts présentés ci-dessus ont été développés par Laurent Oudot
(http ://www.rstack.org/oudot/prelude/correlation/) qui est un des premiers à se pencher sur le sujet.
Il est à noter que la visualisation des vulnérabilités détectées par Nessus n'est pas possible à partir du frontend de Prelude. Elles sont uniquement ajoutées à la base de données.
Bien que cette nouvelle fonctionnalité soit enthousiasmante, nous pouvons dire que nous sommes encore loin d'une véritable intégration de Nessus dans Prelude-IDS.

6 Prelude-IDS et ses compères

6.1 Prelude-IDS, Honeyd et Systrace

6.1.1 Honeyd

Présentation
Honeyd est un projet "Pot de miel" OpenSource développé par Niels Provos. Il permet de déployer des machines virtuelles sur un réseau (en utilisant les adresses IP laissées libres) et ainsi permet de détecter toute activité frauduleuse sur le réseau. En effet, toute tentative de connexion sur une adresse IP non attribuée peut être considérée comme non autorisée et suspecte puisque n'ayant pas lieu d'être.
Le but est aussi bien de détecter des attaques connues que de découvrir de nouvelles attaques, en observant le comportement des attaquants, qui ne sont pas encore recensées. Ainsi, on souhaite accroître la sécurité du réseau.

Fonctionnement

Honeyd fonctionne sous environnement Unix, Solaris, *BSD, et sera porté dans l'avenir sous Windows. C'est un daemon qui crée des hosts virtuels sur le réseau utilisant les adresses IP non attribuées sur le réseau. Ces hosts peuvent être configurés pour qu'ils paraissent fonctionner sous certains systèmes d'exploitations (grâce à des templates) ou pour faire tourner certains services (en fait ce sont des scripts qui simulent le fonctionnement de ces services).
C'est un pot de miel basse interaction : simule des services TCP/IP, peut avoir plusieurs adresses IP (jusqu'à 65536 testées) et supporte ICMP (les machines virtuelles répondent aux ping et aux traceroute). Toutes les données entrantes et sortantes (chaque paquet passe dans un "moteur personnalisé" qui permet de compléter les paquets avec les informations relatives au système d'exploitation simulé) du pot de miel sont analysées.

Plus précisement, Honeyd doit être utilisé en collaboration avec l'outil Arpd. Arpd permet de gérer les adresses IP non attribuées et il redirige les attaques vers Honeyd. Quand à lui, Honeyd gère les échanges de données avec les attaquants pour simuler les services, requêtes ICMP...
Sans Arpd, Honeyd ne peut pas travailler.

Il existe tout de même des limitations à l'utilisation de Honeyd puisque peu de services simulés sont disponibles et qu'il ne simule pas tous les systèmes d'exploitation.

Comparaison entre Honeyd et Prélude
Prelude remonte des alertes en fonction des signatures d'attaques que nous lui fournissons alors que Honeyd remonte toutes les données qui lui sont envoyées (puisque celles-ci sont considérées comme toutes suspectes).

Intégration de Honeyd dans Prélude
Honeyd peut être intégré dans l'architecture Prelude-IDS en tant que sonde réseau et ainsi on peut disposer d'un "Pot de miel" afin d'obtenir de nouvelles informations sur ceux qui attaquent le réseau. Les alertes sont visibles à partir du frontend de Prelude-IDS.
Pour cela, il suffit d'appliquer un patch à la libprelude et la sonde Honeyd pourra être manipulée comme n'importe quelle autre sonde Prelude-IDS. Tout cela est expliqué en détail en annexe.

6.1.2 Systrace

Présentation
Systrace est un outil pour controler de manière très détaillée le comportement et les droits des applications au niveau appel système.
Pour chaque programme exécuté sur une machine, Systrace permet de définir un ensemble d'appels système (system calls) que le logiciel a le droit ou non d'exécuter : ouverture/écriture de fichiers, lecture/écriture d'une socket TCP/IP, allocation de mémoire, ... L'ensemble de ces règles est appelé police. Cette police est générée de manière interactive et les accès non définis génèrent une alarme.

Fonctionnement
Tout d'abord, les polices sont contenues dans un fichier qui a pour nom le chemin de l'application protégée (en remplaçant les / par des _). Par exemple, la police pour l'application /usr/sbin/named est contenue dans le fichier usr_sbin_named.
Dans ces fichiers, on trouve des restrictions d'acces pour les appels système.
Exemples :

```
$--$ native-accept: permit

L'appel système accept() est autorisé pour tout le monde.

$--$ native-bind: sockaddr match "inet-*:53" then permit
```

Lors de l'appel de la primitive bind(), seuls les appels dont sockaddr correspond à la chaine de caractères `inet-* :53` sont autorisés.

Intégration de Systrace dans Prelude

Systrace est surtout nécessaire à Prelude dans le cas où on utilise Honeyd. En effet, comme Honeyd échange des informations avec des attaquants potentiels, il faut protéger le système en évitant que ces attaquants n'exploitent les failles de Honeyd : c'est le rôle de Systrace.
L'intégration se fait uniquement en installant Systrace et en appliquant un patch.
Systrace devient alors une sonde Prelude comme une autre.
Le patch est encore en développement donc nous ne l'avons pas testé.

Installation

Nous n'avons pas installé Systrace. Néanmoins, c'est un package de la Debian (en unstable et testing seuleument) : systrace1.0-4
Le patch se trouve à l'adresse suivante :
`http ://www.rstack.org/oudot/prelude/systrace/files/`

7 Utilisation de Prelude (positionnement dans un réseau)

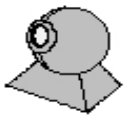
Nous trouvons sur Internet des recettes toutes faites pour positionner les sondes sur un réseau, quelque soit les besoins. Il serait faux de penser que tous les réseaux doivent être protégés de la même manière.
Nous allons ici présenter 4 solutions :

– Suivi des attaques : Les sondes Prelude-IDS sont positionnées sur tout le réseau afin de suivre le chemin parcourus par une attaque.
– Surveillance simple : Les sondes sont placées afin d'avoir qu'une seule remontée d'alerte en cas de problème sur un seul réseau. Surveillance de tout le réseau mais maintenance moins lourde.
– Surveillance ciblée : Comme précédemment si ce n'est qu'ici, seuls les réseaux sensibles sont surveillés.
– Surveillance multi-sites, multi-responsabilités : Dans le cas d'un réseau milti-sites, nous avons étudié le cas de gestion des alertes par plusieurs managers.

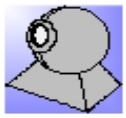
Mais bien que ces scénarios soient beaucoup plus élaborés que ce que l'on nous donne d'habitude, il faudra garder à l'esprit que le meilleur scénario est celui qui sera propre à notre réseau.
Avant d'aller plus loin, voici la légende utilisée dans les schémas :

Réseau

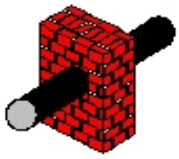
Prelude-manager



Sonde NIDS



Sonde HIDS



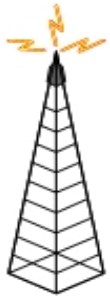
Firewall avec tunneling

Routeur

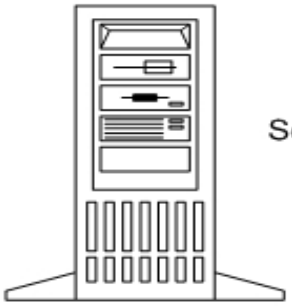
Firewall

Poste de travail nomade

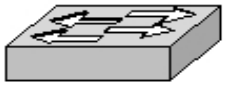
Réseau local ethernet



Access point wireless



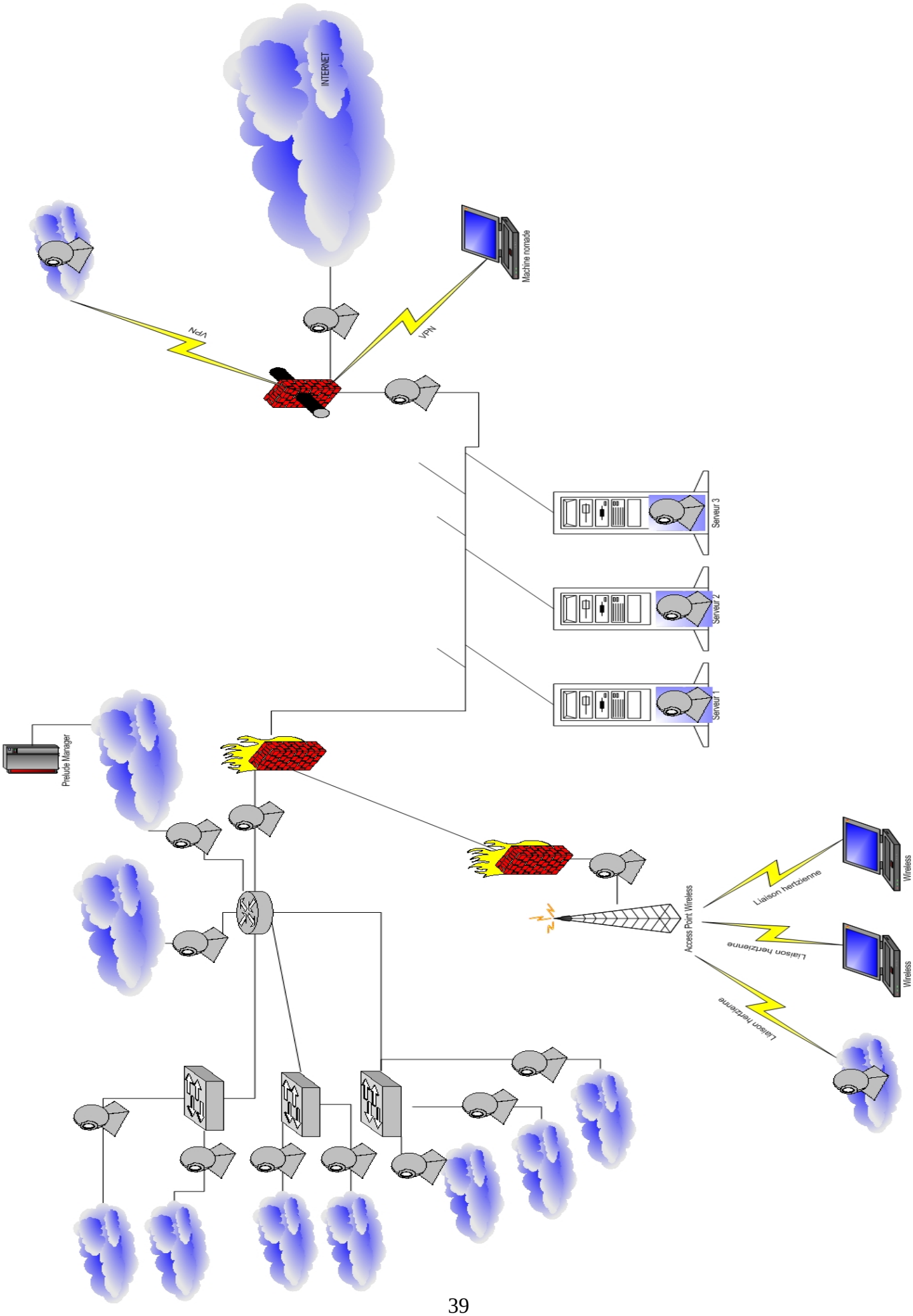
Serveur



Commutateur

Liaison particulière (VPN, sans-fil)

7.1 Suivi des attaques



Les IDS sont rarements les seules solutions de sécurité informatiques mises en place sur un réseau. Il y a donc tout un ensemble de “barrières” déjà mises en place. Mais sont-elles efficaces ? Et si une attaque est perpétrée contre une machine du réseau, quelle protection a permis de la stopper, et laquelle n’a au contraire rien changé ? Un réseau peut aussi avoir des vulnérabilités de conception, comme par exemple, la possibilité de se connecter à des Access Point Wireless, accédant de la sorte au LAN sans passer par le firewall qui ne surveille que les accès à Internet.

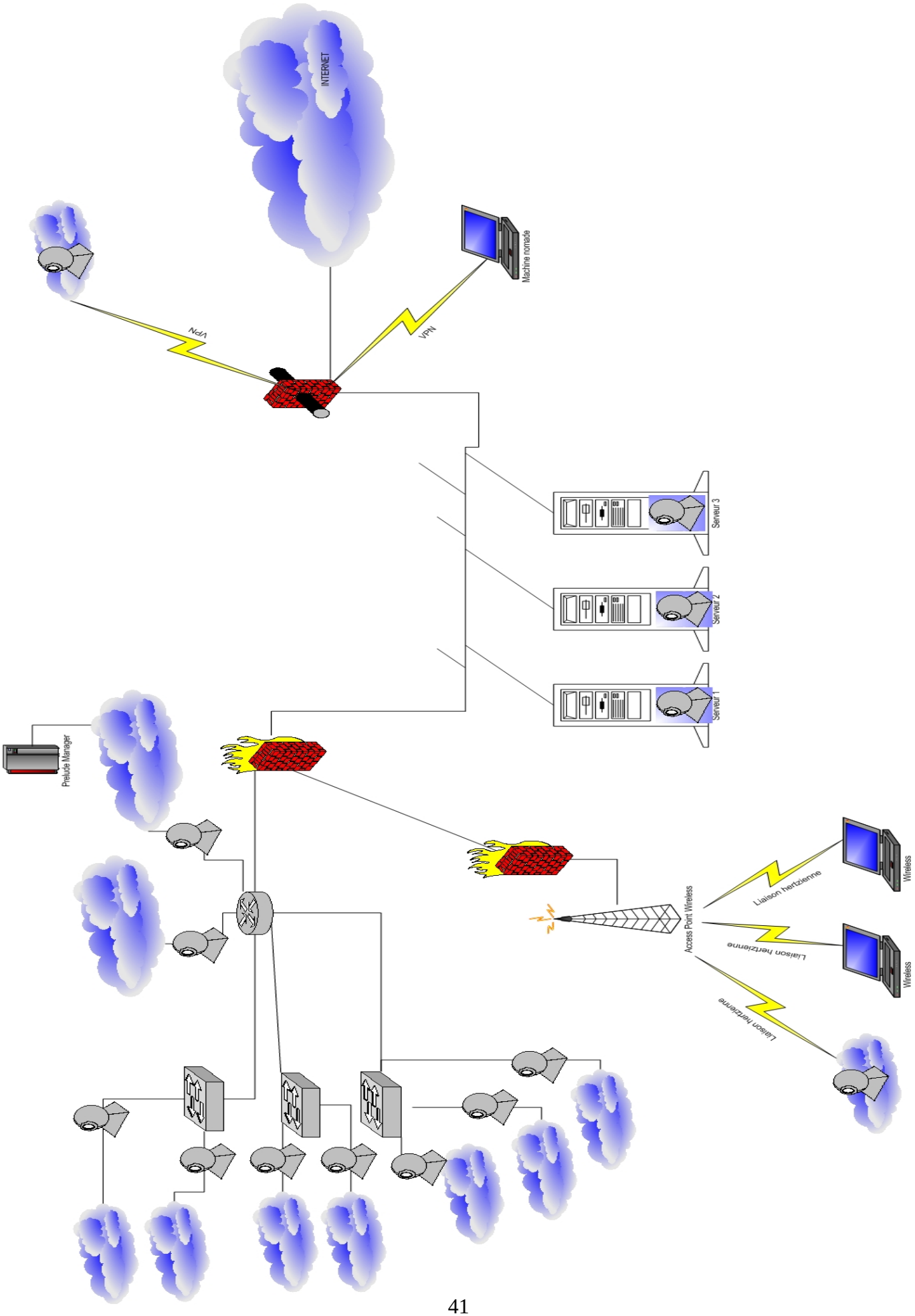
La mise en place de sondes sur tout le réseau va permettre de suivre une attaque. Il est important de pouvoir avoir une sonde surveillant le traffic avant et après chaque protection (avant et après un firewall, ...). Ainsi, cela permet d’avoir un aperçu des attaques que la protection bloque et celles qu’elle laisse passer. Pour une attaque donnée, on pourra en déduire son *taux de pénétration* dans le réseau, et ainsi désigner les parties du réseau sûres et celles non sûres face à une attaque donnée.

Dans les réseaux complexes et vastes, nous nous trouvons parfois face à des niveaux de sécurité hétérogènes. Le cas le plus répandu est de rencontrer une sécurité maximale sur l’accès à Internet, mais une fois dans le LAN, tout est ouvert (faible protection entre la DMZ et le LAN, des Access Point Wireless poussant comme des champignons, accèss illicites à Internet directement d’un poste de travail, ...). Alors qu’une attaque donnée serait bloquée en passant par la grande porte, elle ne rencontrera aucune résistance en passant par un autre chemin. Le positionnement des sondes sur tout le réseau va alors permettre de détecter par où elle est rentrée, ou le chemin qu’elle a empruntée pour contourner les protections qui auraient pu la bloquer.

Un dernier point : une attaque par rebond s’appuie sur une première connexion plus ou moins autorisée à une machine qui va ensuite servir d’attaquant intermédiaire. Toujours avec le positionnement des sondes décrit ici, nous pourrons voir la source de l’attaque (attaquant intermédiaire) et plus seulement son point d’entrée.

Avantages/Inconvénients :
Le problème de cette solution est le coût matériel et humain. Le nombre de sondes étant très important, il est évident que le nombre de machines à dédier à cette tâche est plus important. De plus, l’exploitation correcte de toutes les alertes remontées (alertes remontées par plusieurs sondes, savoir retrouver le parcours d’une attaque dans tout cela, ...) nécessite beaucoup de temps.

7.2 Surveillance simple

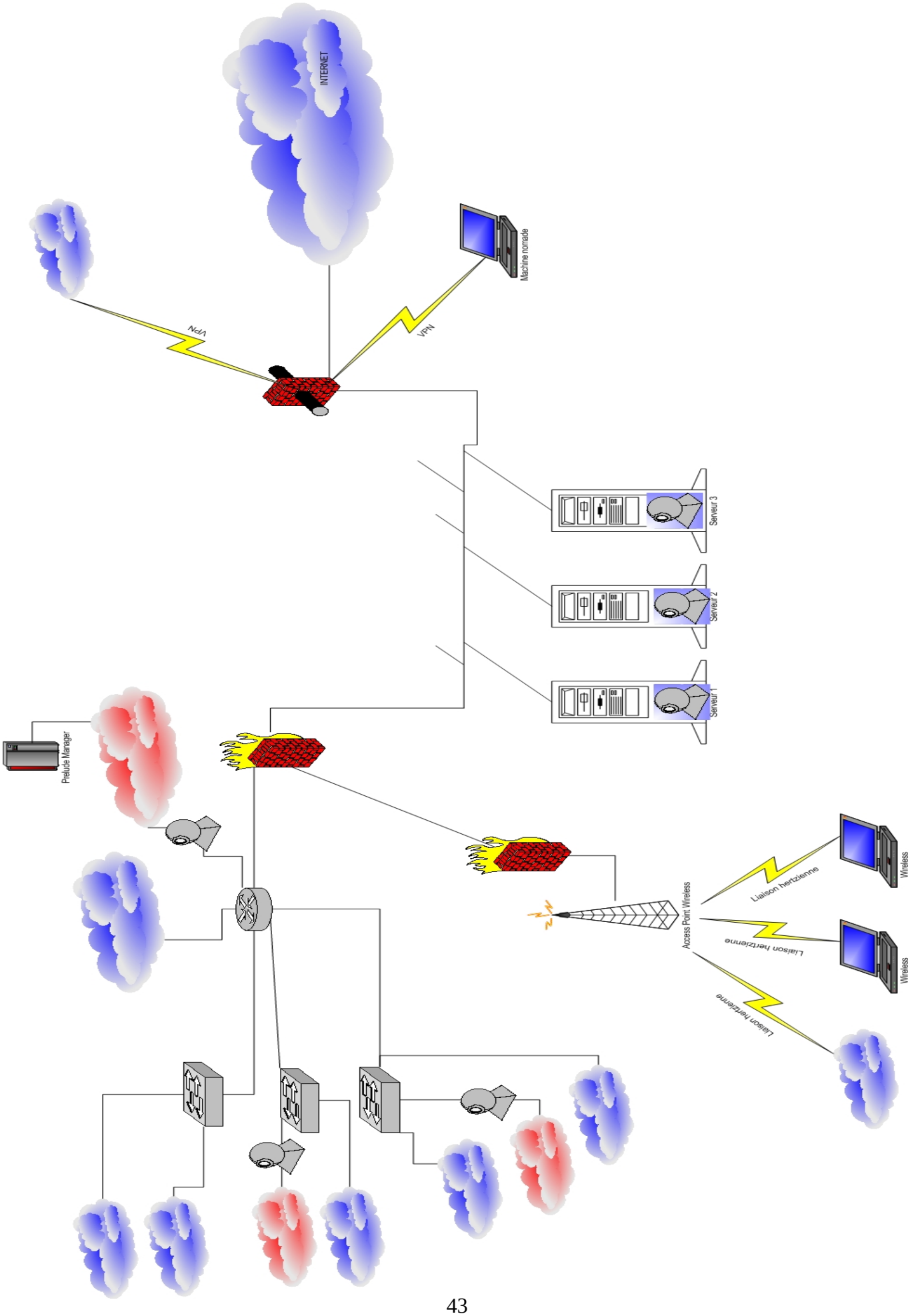


Il est ici question de détecter une attaque. Le chemin suivi par cette dernière n'est plus important. Les sondes sont placées seulement sur les points d'entrées des réseaux et absentes du *backbone*.

Alors que précédemment les sondes devaient toutes avoir activées les mêmes signatures d'attaques pour le suivi, nous pouvons ici faire un paramétrage plus fin. Dans un réseau A sous Windows (IIS), on va activer les alertes unicode, mais sous un réseau B, non. Nous évitons de la sorte les remontées d'alertes pour des attaques dont la cible est invulnérable. L'administration des IDS s'en trouve tout autant simplifiée.

Avantages/Inconvénients :
Contrairement au cas précédent, le nombre de sondes est beaucoup plus réduit. Étant donné que nous recherchons à détecter les attaques dont nous sommes vulnérable, nous pouvons alléger les remontées d'alertes. Cette solution est alors beaucoup moins coûteuse que la précédente. Le revers de la médaille est que l'on perd en précision, et qu'il n'y a plus de redondance dans la surveillance (zone surveillée par au moins deux sondes).

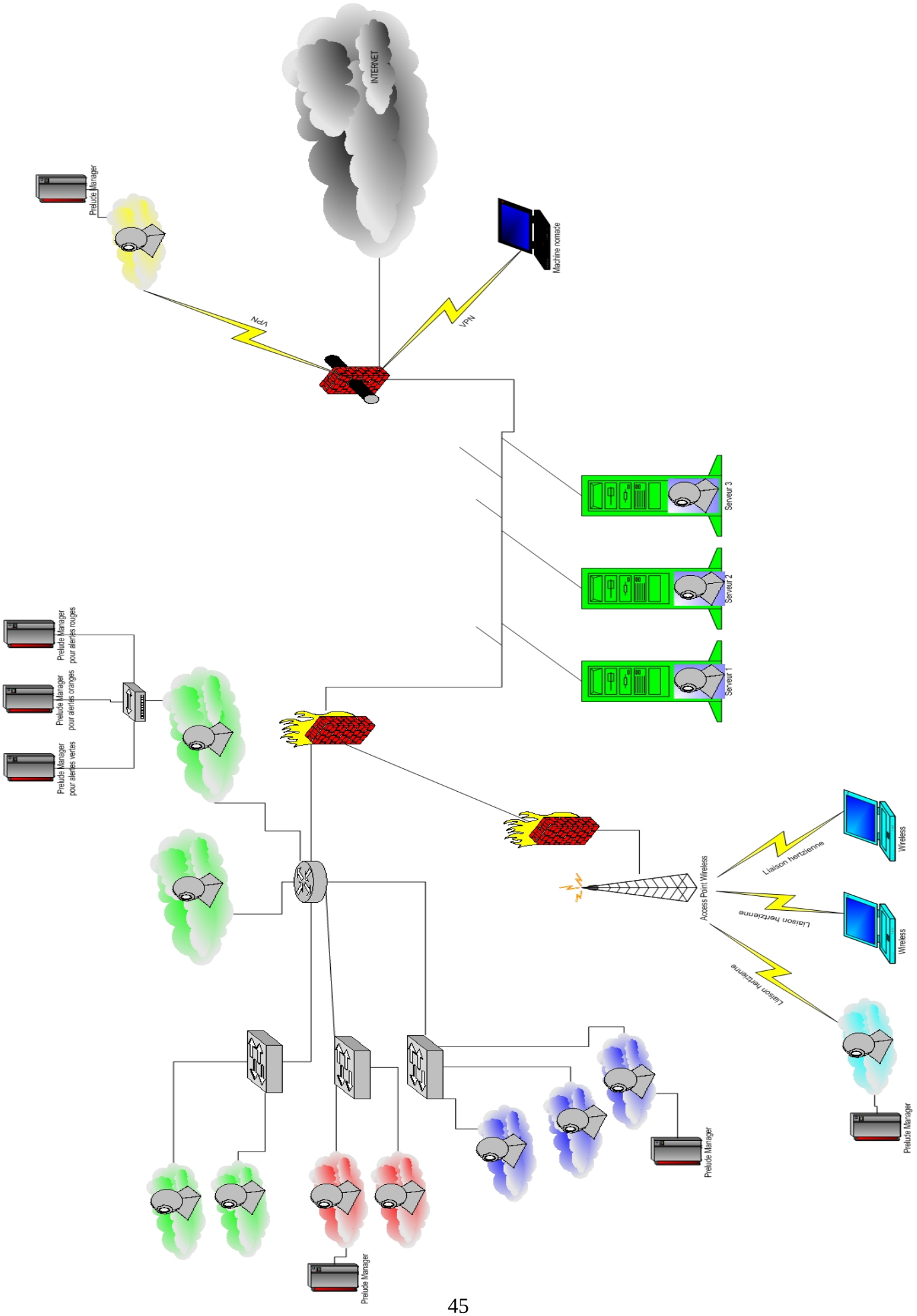
7.3 Surveillance ciblée



Jusqu'à présent, le nombre de sondes est important. Cela implique une maintenance lourde et coûteuse sur les grands réseaux. Les points d'un réseau ne sont pas aussi sensibles les uns que les autres. Cela est valable sur les réseaux très vaste : il est préférable de bien surveiller les parties sensibles du réseau que de vouloir tout contrôler avec le même niveau de sécurité ... chose qui est impossible (ou alors chère).
La mise en place doit, bien sûr, être précédée d'une phase d'étude pour déterminer quelles sont les zones sensibles, et celles qui le sont moins.

Avantages/Inconvénients :
Le coût est ici très faible. La complexité de la surveillance est réduite, ce qui fait gagner d'autant en efficacité et en rapidité de réaction en cas d'incident de sécurité. Nous comprenons bien que l'efficacité des IDS dépendra beaucoup de l'étude préalable du réseau.

7.4 Surveillance multi-sites, multi-responsabilités



Plusieurs sites distincts : vert, rouge, bleu, cyan, jaune, et noir pour l'extérieur.

Nous traitons ici le cas particulier des réseaux très étendus, multi-sites, dont la gestion de la sécurité n'est pas entièrement centralisée. Si un incident de sécurité se produit sur le site A, ce serait une perte de temps de remonter l'alerte sur le site C où un responsable sécurité préviendra le responsable sécurité du site A de l'incident. Alors autant faire en sorte que les alertes de chaque sonde soit remontées au manager du site.

La modularité des filtres sur les managers permettent même d'aller plus loin. En effet, rien ne nous empêche de positionner plusieurs managers comme dans le site vert, afin qu'un premier manager n'affiche que les alertes vertes, un autre que les alertes oranges, et ainsi de suite. Pour le moment cette gestion ne peut pas se faire sur les sondes, mais c'est actuellement en discussion sur la mailling liste de Prelude (nous avons soumis l'idée et les développeurs semblent intéressés mais la question est de savoir comment faire, et si cela ne serait pas trop difficile à implémenter). Ce partage entre plusieurs managers permet de rediriger les alertes vers différentes personnes, suivant leurs niveaux de compétences. Nous pourrions rediriger les alertes vertes vers un technicien, les alertes oranges vers un ingénieur réseau, et les alertes rouges vers un ingénieur expert sécurité. Nous reproduirions alors les niveaux supports, au niveau de la maintenance sécurité : support de niveau 1, puis si le problème persiste nous passons au niveau 2, etc. Ici, il y aurait en plus la possibilité pour une alerte d'arriver directement au niveau 2 ou 3, suivant sa gravité (réseau très sensible à ce type d'attaque, alertes rouges, ...).

Enfin, une dernière possibilité serait de combiner le traitement des alertes par un manager sur chaque site mais, dans le même temps, remonter les alertes vers un autre manager qui centraliserait ainsi les alertes de TOUT le réseau informatique de l'organisme (pour réaliser un historique par exemple).

Avantages/Inconvénients :
Il n'est plus question ici de surveillance d'un petit réseau, mais de l'intégration des IDS dans une politique de sécurité globale de l'organisme. Les traitements séparés avec plusieurs niveaux d'intervention montrent que la sécurité n'est pas que technique mais aussi organisationnelle.
L'avantage de cette solution est donc de s'intégrer totalement dans une politique de sécurité, ce qui induit qu'il est nécessaire d'avoir une organisation de la sécurité stable, et bien définie.

8 Conclusion

Snort et Prelude sont des outils très fiables et très intéressants dans la mise en place d'une sécurité réseau. Cependant les différences de fonctionnement et de gestion sont nombreuses, ce qui rend la comparaison très difficile. Les avantages de l'un pouvant être considérés comme des inconvénients par l'autre (exemple de la réponse active, ou de l'archivage des payloads). De plus, les règles de Snort pouvant être intégrées dans Prelude-IDS, ce n'est plus sur la quantité d'attaques reconnues que l'on pourra comparer les deux outils. Nous pourrons donc en conclure qu'ils sont très proches et que, suivant l'utilisation que l'on voudra en faire, on choisiera soit Snort, soit Prelude-IDS. Notons tout de même que l'interconnexion de Prelude-IDS avec d'autres outils est quand même un avantage considérable sur Snort dans la mise en place d'une solution de sécurité globale.

L'objectif final du TER qu'était la mise en production a finalement été abandonnée. Les problèmes récurrents d'installation des sondes (dus au fait que les machines dédiées étaient obsolètes) nous a coûté beaucoup de temps. Nous avons donc choisi de recentrer nos efforts sur l'étude.
C'est ainsi que nous avons travaillé sur l'intégration de Nessus, de Honeyd, ou encore de Systrace dans Prelude-IDS. De la sorte, nous avons pu proposer des manuels d'installations pour ces "plugins" un peu particuliers, et proposer ainsi une solution sécurité la plus efficace possible.

Beaucoup d'IDS sont matures et fiables, ce qui entraîne leur intégration quasi systématique dans les solutions de sécurité. Les IDS présentent des avantages par rapport aux autres outils, mais aussi des inconvénients, des apports ainsi que des lacunes, qui en font des outils indispensables et non-suffisants.

La particularité des systèmes de détection d'intrusions est d'apporter une sécurité non-automatisée. Nous entendons par cela que la mise en place doit toujours s'accompagner d'une expertise sécurité dans les traitements des alertes remontées.

En ce qui concerne l'environnement d'étude, la particularité de travailler sur un projet libre, alors qu'il se trouve dans une période très active, est très enrichissante. C'était pour nous trois la première fois que nous étions en contact direct avec les développeurs d'un projet si important.

Pour finir, nous dirons que tout ce travail a été très enrichissant techniquement et dans la méthodologie de travail, car il nous a ouvert un peu plus les portes de la sécurité informatique, et plus particulièrement sur les IDS. Cette vaste étude nous permet d'affirmer que nous possédons maintenant de bonnes compétences sur les Systèmes de Détection d'Intrusion ... et que Prelude-IDS et Snort sont tous deux de bons outils avec leur propres particularités.

9 Bibliographie

9.1 Installation et configuration de Prelude-IDS

Site officiel de Prelude-IDS : http ://www.prelude-ids.org

Dernière version du document d'installation de Prelude-IDS : http ://lehmann.free.fr

Téléchargement des paquetages de Prelude-IDS :

http ://www.prelude-ids.org/rubrique.php3 ?id_rubrique=6

Autres documentations à propos de Prelude-IDS :

http ://www.prelude-ids.org/rubrique.php3 ?id_rubrique=1

Site officiel du frontend perl : http ://www.leroutier.net/Projects/

Site de Debian : http ://www.debian.org

Se procurer Debian sur CD : http ://ikarios.com/form/

9.2 Aide sur les Systèmes de Détection d'Intrusion

http ://www.securiteinfo.com : Site sur la sécurité informatique. On peut y trouver des explications intéressantes sur les différentes attaques existantes.

http ://www.snort.org/docs/idspaper/ : Document présentant les faiblesses des IDS.

http ://www.secusys.com : Site sur la sécurité informatique. Contient quelques documents intéressant lors d'un tout premier contact avec ce domaine.

Misc Num 3 : Magazine sur la sécurité informatique. Le numéro 3 contient un dossier spécial sur les IDS.

www.security-labs.org/index.php3 ?page=408

www.securiteinfo.com/conseils/choix_ids.shtml

www.securite.org/db/securite/ids/outils

http ://abcdelasecurite.free.fr/html/modules.php ?name=Downloads

9.3 Bibliographie pour intégrer les règles de Snort à Prelude-NIDS

Site officiel de Prelude-IDS : http ://www.prelude-ids.org

Site officiel de snort : http ://www.snort.org

Site où l'on peut télécharger le script convert_ruleset :
http ://mops.uci.agh.edu.pl/ kzaraska/prelude-contribs/

Téléchargement des paquetages de Prelude-IDS :
http ://www.prelude-ids.org/rubrique.php3 ?id_rubrique=6

Téléchargement des règles de Snort : http ://www.snort.org/dl/rules/

9.4 Projets liés de Prelude-IDS

Site officiel du projet Trithème : http ://tritheme.sourceforge.net/

Bibliographie sur Honeyd

http ://www.citi.umich.edu/u/provos/honeyd/

http ://www.citi.umich.edu/u/provos/papers/honeyd-dfn/
http ://www.securityfocus.com/infocus/1659
http ://www.citi.umich.edu/u/provos/systrace/
http ://www.onlamp.com/pub/a/bsd/2003/01/30/Big_Scary_Daemons.html

Bibliographie sur Nessus

Site officiel de Nessus : http ://www.nessus.org
Corrélation entre Prelude et Nessus : http ://www.rstack.org/oudot/prelude/correlation/

Bibliographie sur Logcheck

Fiche sur logcheck : http ://www.micro-fun.ch/techniques/fiches/controle_logs.shtml

10 Licence

Copyright (C) 2003 Guillaume LEHMANN, Dado KONTAE, Clément LORVAO Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation ; with no Invariant Sections, with no Front-Cover Texts, and one Back-Cover Text :
“La version originale de ce document a été publiée par Guillaume LEHMANN, Dado KONATE, et Clément LORVAO dans le cadre d’un Travail d’Étude et de Recherche en DESS STRI. Pour plus de renseignements : http ://lehmann.free.fr/ ou écrivez à lehmann@free.fr”.

A Insertion des règles de Snort dans Prelude-NIDS

Ce document explique comment intégrer les règles de Snort sous Prelude-IDS. Cela ne concerne biensûr que la sonde NIDS.
La plateforme de travail est une Debian woody, et nous utilisons les versions suivantes des sources de Prelude-IDS : libprelude-0.8.4, prelude-nids-0.8.1 .
Nous considérons que la sonde Prelude-NIDS est déjà correctement installé et configurée.

A.1 Intégration des règles

Afin d'avoir le fichier de configuration de snort, il faudra l'installer :
`apt-get install snort`
Ensuite, télécharger les règles de Snort. Nous les décompresserons dans le répertoire `/home/user/` .
Apparaît alors le répertoire `rules` contenant toutes les règles.

Donner les droits en exécution au script et l'exécuter de la façon suivante :
`./convert_ruleset /etc/snort/ /home/user/rules/ /usr/local/etc/prelude-nids/`

Ca y est, les nouvelles règles sont maitenant installées.

A.2 Le code source du script convert_ruleset

```
#!/bin/sh

if test -z $1 || test -z $2 || test -z $3; then
echo "This script convert Snort ruleset to Prelude ruleset"
echo
echo "$0 <Snort configdir> <Snort ruledir> <Prelude ruledir>"
exit 1;
fi

snort_confdir=$1
snort_ruledir=$2
prelude_ruledir=$3

grep ^var $snort_confdir/snort.conf > $prelude_ruledir/prelude.rules

echo "" >> $prelude_ruledir/prelude.rules
echo "" >> $prelude_ruledir/prelude.rules

grep include $snort_confdir/snort.conf >> $prelude_ruledir/prelude.rules

cp $snort_ruledir/*.rules $prelude_ruledir

for i in $snort_confdir/*.config; do

if [[ "$i" == "classification.config" ]]; then
continue;
fi
```

```
        cp $i $prelude_ruledir
done
```

A.3 Le code source du script convert_ruleset commenté

```
#!/bin/sh

####
# Parametres rantres ensuite, donc pas lors de l'execution de la commande.
#if test -z $1 || test -z $2 || test -z $3; then
# echo "This script convert Snort ruleset to Prelude ruleset"
# echo
# echo "$0 <Snort configdir> <Snort ruledir> <Prelude
ruledir>"
# exit 1;
#fi

snort_confdir=$1    # Dans ce qui suit, snort_confdir est devenu /etc/snort
snort_ruledir=$2    # Dans ce qui suit, snort_ruledir est devenu rules
prelude_ruledir=$3  # Dans ce qui suit, prelude_ruledir est
                    #devenu /usr/local/etc/prelude-nids

####
# Ecraser la conf pr?sente dans prelude.rules, par celle du fichier de conf d
snort.conf .
grep ^var /etc/snort/snort.conf > /usr/local/etc/prelude-nids/prelude.rules

####
# Rajout d'un saut de 2 lignes dans le fichier listant les types de règles
utilisées.
echo "" >> /usr/local/etc/prelude-nids/prelude.rules
echo "" >> /usr/local/etc/prelude-nids/prelude.rules

####
# Ajout de la liste des règles de snort.conf à celle de prelude.rules
grep include /etc/snort/snort.conf >>
/usr/local/etc/prelude-nids/prelude.rules

####
# Copie des nouvelles regles de snort dans le repertoire de prelude approprie
cp rules/*.rules /usr/local/etc/prelude-nids

####
# Normalement, il n'y a que le fichier classification.config qui est concerne
La boucle "for" est ici pour parcourir le répertoire a la recherche de
ce fichier classification.config . On travaille donc sur le fichier qui d?fin
le niveau des alertes. On copie ce fichier dans le répertoire de prelude.
for i in /etc/snort/*.config; do
```

```
if [[ "$i" == "classification.config" ]]; then
continue;
fi

        cp $i /usr/local/etc/prelude-nids/
done
```

B Installation de Honeyd avec Prelude-IDS

1) Installation de la *libprelude* sur le système :
Voir la documentation sur l'installation de Prelude-IDS en annexe.

2) On applique le patch aux sources de Honeyd (http ://www.rstack.org/oudot/prelude/) :

```
# patch -p0 < honeyd-as-a-prelude-ids-sensor.patch
patching file honeyd-0.5/Makefile.am
patching file honeyd-0.5/Makefile.in
patching file honeyd-0.5/README.prelude-ids
patching file honeyd-0.5/command.c
patching file honeyd-0.5/honeyd.c
patching file honeyd-0.5/prelude_alert.c
patching file honeyd-0.5/prelude_alert.h
#
```

3) On configure puis on installe Honeyd :
Cette installation nécessite les librairies libevent, libdnet et libpcap.

```
# ./configure
# make
# make install
```

4) On déclare Honeyd comme une sonde sur le Prelude Manager :

```
# sensor-adduser --sensorname "honeyd" --uid 0 -m 130.120.84.63
```

5) On modifie la configuration de Prelude dans /usr/local/etc/prelude-sensor/sensor-default.conf :

```
manager-addr : 130.120.84.63
```

6) On installe Arpd (arpd-0.2.tar.gz) :

```
# ./configure
# make
# make install
```

7) On lance Arpd :

```
# ./arpd 130.120.84.0/24
arpd[13817]: listening on eth0: arp and (dst net 130.120.84.0/24) and not

ether src 00:10:5a:d8:8b:51
#
```

A partir de maintenant, le daemon Arpd gère toutes les adresses IP libres sur le réseau 130.120.84.0/24

8) On lance Honeyd :

```
# honeyd -d -p nmap.prints -f config.sample 130.120.84.0/24 -l honeyd.log
- Connecting to Tcp prelude Manager server 130.120.84.63:5554.

- SSL authentication succeed with Prelude Manager.

honeyd[13826]: listening on eth0: ip and (dst net 130.120.84.0/24) and

not ether src 00:10:5a:d8:8b:51

honeyd[13826]: Sending ICMP Echo Reply: 130.120.84.67 -> 130.120.84.63
honeyd[13826]: Sending ICMP Echo Reply: 130.120.84.67 -> 130.120.84.63
honeyd[13826]: Sending ICMP Echo Reply: 130.120.84.67 -> 130.120.84.63
honeyd[13826]: Sending ICMP Echo Reply: 130.120.84.67 -> 130.120.84.63
honeyd[13826]: Sending ICMP Echo Reply: 130.120.84.67 -> 130.120.84.63
#
```

On peut voir ici que Honeyd a répondu à une requête ICMP. La machine qui a initié cette réponse (130.120.84.63) s'est adressé à une machine qui n'existe pas sur le réseau (130.120.84.67). Une alerte est donc générée et envoyée au Prelude-manager. Nous pouvons donc visualiser cette alerte sur le frontend de Prelude.

C Installation de Nessus et corrélation avec Prelude-IDS

C.1 Installation de Nessus

1) L'installation de Nessus requiert deux packages :
Installation du serveur Nessus :

```
#apt-get install nessusd
```

Installation du client Nessus :

```
#apt-get install nessus
```

2) Ajout d'un utilisateur et définition de la politique de sécurité appliquée :

```
manager:/# nessus-adduser

Add a new nessusd user
----------------------

Login : prenessus
Authentication method (cipher/plaintext) [cipher] :

Source restriction
------------------

Source host or network [anywhere] : localhost
One time password : -stri-

User rules
----------
nessusd has a rules system which allows you to restrict the hosts
that prenessus has the right to test. For instance, you may want
him to be able to scan his own host only.

Enter the rules for this user, and hit ctrl-D once you are done :
(the user can have an empty rules set)

default allow

Login             : prenessus
Auth. method      : cipher, can connect from localhost
One time password : -stri-
Rules             : default allow

Is that ok ? (y/n) [y] y
user added.

manager:/#
```

3) Lancer le serveur nessusd en mode démon (option -D) :

```
manager:/# nessusd -D
```

4) Lancer et configurer le client nessus :

```
manager:/#nessus
```

C.2 Intégration de Nessus dans Prelude

Il est possible d'intégrer les rapports de Nessus dans la base de données de Prelude. Pour cela, il faut : Créer la table de corrélation Nessus/Prelude dans la base de données Prelude :

```
mysql --database prelude < prelude-correlation-vulne
rability-importer_mysql.sql
```

Ensuite, intégrer le rapport de Nessus dans la base de données Prelude avec la commande suivante :

```
manager:/home/prelude# ./prelude-correlation-vulner
ability-importer.pl -c vuln.conf
-i 130_120_84_65.nsr
***
Trying to connect to the database
 type:mysql      database:prelude
 hostname:localhost      account:prelude
***
Reading the report file: 130_120_84_65.nsr
Number of reports to deal with: 134
***
Deleting existing related reports
.....................................................................
.....................................................
End of the deletion process
***
Adding 134 new reports
End of the integration of the reports
***
Operation of import succeeded

 Number of records deleted in the database : 134
 Number of records written in the database : 134
 Date used in the records                  : 2003/2 3 11:30

manager:/home/prelude#
```

Afin de faire la corrélation entre les alertes relevées par Prelude et les vulnérabilités trouvées par Nessus, on utilise le script suivant :

```
manager:/home/prelude# ./prelude-correlation-vulner
ability-finder.pl -B -c vuln.conf
*** Trying to connect to the database
 type:mysql      database:prelude
 hostname:localhost      account:prelude

*** Parsing the database to get the alerts id

*** Trying to search for correlation
*** Operation of vulnerability correlation search finnished

 Number of alerts treated in the database : 1073
 Number of correlations of vulnerability  : 0
manager:/home/prelude#
```

Ici on peut voir qu'il n'y a eu aucune attaque exploitant une vulnérabilité préalablement détectée par Nessus.
Il existe aussi un script très simpliste pour faire un tri sur les corrélations durant un laps de temps donné. Par exemple, pour afficher les corrélations pour les dernières 24 heures (ou les 86400 dernières secondes) :

```
manager:/home/prelude/# ./prelude-correlation-agent.pl -B -p 86400
##################################################
# Proof of concept for a small correlation agent
# using source address to work
# Contact: oudot@rstack.org
##################################################


# Source address correlation for alert 15972
# with 86400 of seconds used as the period to analyse

# Alert 15972 is ever correlated in the correlation id: 1

##################################################
# Proof of concept for a small correlation agent
# using source address to work
# Contact: oudot@rstack.org
##################################################


# Source address correlation for alert 15973
# with 86400 of seconds used as the period to analyse

# Alert 15973 is ever correlated in the correlation id: 1


[...]


##################################################
# Proof of concept for a small correlation agent
# using source address to work
# Contact: oudot@rstack.org
##################################################


# Source address correlation for alert 17043
# with 86400 of seconds used as the period to analyse

# Alert 17043 is ever correlated in the correlation id: 37

##################################################
# Proof of concept for a small correlation agent
# using source address to work
# Contact: oudot@rstack.org
##################################################
```

```
# Source address correlation for alert 17044
# with 86400 of seconds used as the period to analyse

# Alert 17044 is ever correlated in the correlation id: 37
```

D Manuel d'installation de Prelude-lml/NIDS/manager, et des frontends perl et php

D.1 Introduction

Nous expliquons l'installation et la configuration de Prelude-IDS et de tous les paquetages qu'il utilise. La plateforme de travail est une Debian 3.0 en testing, et nous utilisons les versions suivantes des sources de Prelude-IDS : libprelude-0.8.4, prelude-manager-0.8.6, prelude-nids-0.8.1, et prelude-lml-0.8.2 . Nous avons choisit de stocker les alertes reçues par le manager dans une base de donnée MySQL. L'installation et la configuration de cette dernière sera donc décrite dans ce document.

D.2 Paquetages nécessaires

Il est nécessaire de télécharger, sur le site www.prelude-ids.org, les paquetages suivants :
– libprelude-x.x.x.tar.gz (nous utiliserons ici la version 0.8.4) ;
– prelude-manager-x.x.x.tar.gz (nous utiliserons ici la version 0.8.6) ;
– prelude-nids-x.x.x.tar.gz (nous utiliserons ici la version 0.8.1) ;
– prelude-lml-x.x.x.tar.gz (nous utiliseront ici la version 0.8.2) ;

libprelude est nécessaire à prelude-nids, prelude-lml, et prelude-manager.

Décompressez les fichiers dans `/usr/local/src/` . Nous avons maintenant les nouveaux répertoire `libprelude-0.8.4, prelude-lml-0.8.2, prelude-manager-0.8.6, et prelude-nids-0.8.1` .

D.3 Installation de paquetages au préalable

Prelude-IDS nécessite pour fonctionner, des bibliothèques et autres applications que nous allons installer maintenant.

D.3.1 Paquetages nécessaires à Prelude-IDS

– **gtk-doc-tools** nécessaire pour libprelude, donc aussi à prelude-nids, prelude-lml, et prelude-manager
– **libssl-dev** nécessaire pour libprelude, donc aussi à prelude-nids, prelude-lml, et prelude-manager
– **mysql-server** à installer sur le poste hébergeant le manager, si l'on veut stocker les alertes dans une bases MySQL
– **libmysqlclient10-dev** nécessaire à prelude-manager
– **libxml2-dev** nécessaire à prelude-manager
– **libpcre3-dev** nécessaire pour prelude-lml
– **libfam-dev** nécessaire pour prelude-lml

D.3.2 Installation de ces paquetages

```
apt-get install gtk-doc-tools
apt-get install libssl-dev
apt-get install mysql-server
```
Pendant l'installation, il est proposé de démarrer le serveur MySQL au démarrage de la machine. Il est conseillé de répondre `Yes`.
```
apt-get install libmysqlclient10-dev
apt-get install libxml2-dev
apt-get install libpcre3-dev
apt-get install libfam-dev
```

D.4 Installation

D.4.1 Installation de libprelude

Rentrez dans le répertoire `/usr/local/src/libprelude-0.8.4/` et tapez les commandes suivantes :

```
./configure --enable-gtk-doc --enable-openssl
make
make install
```

D.4.2 Installation du manager

Rentrez dans le répertoire `/usr/local/src/prelude-manager-0.8.6/` et tapez les commandes suivantes :

```
./configure --enable-gtk-doc --enable-mysql --enable-openssl
```

D.4.3 Installation de la sonde réseau (nids)

Rentrez dans le répertoire `/usr/local/src/prelude-nids-0.8.1/` et tapez les commandes suivantes :

```
./configure --enable-gtk-doc
make
make install
```

D.4.4 Installation de la sonde hôte (lml)

Rentrez dans le répertoire `/usr/local/src/prelude-lml-0.8.2/` et tapez les commandes suivantes :

```
./configure --enable-gtk-doc --enable-fam
make
make install
```

D.5 Configuration

Nous allons aborder ici la configuration de tous les éléments constituant Prelude-IDS. Certaines valeurs comme les adresses IP ou encore les mots-de-passe, seront à adapter par chacun.

D.5.1 Configuration de MySQL

Lancer le démon mysqld si cela n'a pas encore été fait :
`/etc/init.d/mysql restart`
Rentrez dans mysql :
`mysql`

Créez la base de données `prelude`, puis resortir :

```
create database prelude;
quit
```

Donner les droits en exécution (550 par exemple) au fichier `/usr/local/src/prelude-manager-0.8.6/prelude-manager-db-create.sh` comme cela :

Exécutez maintenant le script :
`/usr/local/src/prelude-manager/prelude-manager-db-create.sh`
Il y a 6 phases (de 0 à 5). La première, répondre `y`. À la deuxième, répondre `mysql`. À la troisième, répondre `prelude` (choix par défaut). Pour la quatrième phase qui est la numéro 3, indiquez `root` (choix par défaut) pour l'administrateur de la base, et laisser le mot-de-passe vide car si cela n'a pas été changé, l'utilisateur `root` n'a pas de mot-de-passe par défaut (cela sera à changer ultérieurement pour améliorer la sécurité du système). L'avant-dernière phase attend la réponse `prelude` (choix par défaut), avec `dessstri` comme mot-de-passe. Enfin, pour la dernière phase, répondre `yes` si toutes les informations saisies sont correctes.

D.5.2 Configuration du manager

Éditez le fichier de configuration du manager `/usr/local/etc/prelude-manager/prelude-manager.conf` . Ensuite, faire apparaître les lignes suivantes (les décommenter si elles sont en commentaires, sinon, les écrire dans le fichier) :

Dans la rubrique Prelude Manager :
Indiquer ici l'adresse et le port d'écoute des sondes par le serveur.
#Par défaut, c'est le port 5554 qui est utilisé.
sensors-srvr = 192.168.0.2 ;

Dans la rubrique MySQL :
La ligne ci-dessous est valable même si le manager
#et les sondes ne sont pas sur la même machine.
dbhost = localhost ;
dbname = prelude ;
dbuser = prelude ;
dbpass = dessstri ;

D.5.3 Configuration de la sonde réseau (nids)

Éditez le fichier de configuration de la sonde réseau `/usr/local/etc/prelude-nids/prelude-nids.conf` . Ensuite, faire apparaître les lignes suivantes (les décommenter si elles sont en commentaires, sinon, les écrire dans le fichier) :
`manager-addr = 192.168.0.2 ;`
`user = prelude ;`
Ici, le manager se trouve sur la même machine, sinon, indiquez l'adresse IP de la machine hébergeant le manager. Au cas où le port de communication par défaut serait changé, il faudra l'indiquer à la suite de l'adresse IP, en séparant l'adresse IP et le port de communication de 2 points :
`manager-addr = 192.168.0.2 :5554 ;`

D.5.4 Configuration de la sonde hôte (lml)

Éditez le fichier de configuration de la sonde hôte `/usr/local/etc/prelude-lml/prelude-lml.conf` . Ensuite, faire apparaître les lignes suivantes (les décommenter si elles sont en commentaires, sinon, les écrire dans le fichier) :
`manager-addr = 192.168.0.2 ;`

Ici, le manager se trouve sur la même machine, sinon, indiquez l'adresse IP de la machine hébergeant le manager. Au cas où le port de communication par défaut serait changé, il faudra l'indiquer à la suite de l'adresse IP, en séparant l'adresse IP et le port de communication de 2 points :
`manager-addr = 192.168.0.2 :5554;`

D.6 Lancement de l'écoute

Sur le manager, tapez la commande suivante, dans un terminal, pour créer un utilisateur et avoir un mot-de-passe :
`manager-adduser`
Il sera donné un mot-de-passe à garder.

Sur la sonde, tapez la commande suivant, dans un terminal, pour lancer l'ajout d'un utilisateur dans le sensor :
Pour la sonde réseau, avec 192.168.0.2 pour adresse du manager :
`sensor-adduser -s prelude-nids -m 192.168.0.2 -u 0`

Pour la sonde hôte, avec 192.168.0.2 pour adresse du manager :
`sensor-adduser -s prelude-lml -m 192.168.0.2 -u 0`

Dans un cas comme dans l'autre, la suite est la même. Il est demandé de rentrer le mot-de-passe noté (lors du `manager-adduser`), puis d'indiquer le nom de l'utilisateur (`prelude`) et son mot-de-passe (`dessstri`). Ensuite on accepte de créer cet utilisateur.

On lance ensuite le manager par la commande :
`prelude-manager`

Nous pouvons nous affranchir du fichier de configuration en donnant les paramètres en arguments. Dans notre cas, cela va donner ceci :

```
prelude-manager --mysql --dbhost localhost --dbname prelude --dbuser prelude
 --dbpass dessstri
```

Puis on lance la sonde avec la commande pour la sonde réseau (eth0 est l'interface d'écoute du réseau) :
`prelude-nids -i eth0 -u root`
ou avec la commande suivante pour la sonde hôte :
`prelude-lml -u root`
Il est à noter qu'il faut créer un nouvel utilisateur (`manager-adduser`) pour chaque nouvelle sonde. En revanche, un manager peut écouter, en même temps, les remontées d'alertes de plusieurs sondes.

D.7 Installation et configuration du prelude-php-frontend

D.7.1 Installation

Étant donné que le prelude-php-frontend se base sur un serveur web, nous installerons Apache-ssl, ainsi que php4.
`apt-get install apache-ssl php4 php4-mysql`
Accepter l'ajout de `extension=mysql.so` qui est proposé pendant l'installation.
Ensuite se placer dans le répertoire contenant les sources compressées du prelude-php-frontend, les décompresser, et les mettre dans le répertoire du serveur Apache-ssl :
`cd /usr/local/src/`

```
tar -xzf prelude-php-frontend-0.8.1.tar.gz
```
Est maintenant apparu le répertoire `prelude-php-frontend` que nous copions dans le répertoire du serveur web (`/var/www/`par défaut).
```
cp -r prelude-php-frontend /var/www/
```

D.7.2 Configuration

Éditez le fichier de configuration d'Apache (`/etc/apache-ssl/httpd.conf`), et y écrire où décommenter les lignes suivantes (la machine a pour adresse IP 192.168.0.2) :
```
Listen 192.168.0.2
LoadModule php4_module /usr/lib/apache/1.3/libphp4.so
DirectoryIndex index.php index.html index.htm
ServerName localhost
DocumentRoot /var/www/
```

Éditez le fichier de configuration du prelude-php-frontend (`/var/www/prelude-php-frontend/config.php`), et y écrire où décommenter les lignes suivantes :
```
$server[1]['description'] = "SYSDOOR/MySQL phpfront v".VERSION;
$server[1]['dbtype'] = USE_DB_MYSQL
$server[1]['dbusername'] = "prelude"
$server[1]['dbpassword'] = "dessstri"
$server[1]['dbhostname'] = LOCAL_CONNECTION;
$server[1]['dbport'] = DEFAULT_PORT
$server[1]['dbname'] = "prelude"
```

Éditez le fichier `/var/www/prelude-php-frontend/index.php`, et y modifier la ligne `$serv = 0` en `$serv = 1`

Relancer le serveur web pour prendre en compte les changements :
```
/etc/init.d/apache-ssl restart
```

L'interface est maintenant accessible par *https ://localhost/prelude-php-frontend/*.

ATTENTION : Il y a actuellement un bogue dans le prelude-php-frontend qui fait qu'il est retourné un ligne d'erreur dans l'affichage de la page, lorsque la base de données MySQL est vide. Il suffit donc d'attendre de recevoir quelques alertes avant d'utiliser l'interface.

D.8 Installation et configuration du prelude-perl-frontend

D.8.1 Installation préalable de paquetages

Il faut biensûr que perl et quelques autres modules soient installés. Pour cela, tapez la commande suivante :
```
apt-get install libdbi-perl libgd-graph-perl libdate-calc-perl
apache-ssl
```

D.8.2 Installation de prelude-perl-frontend

Télécharger les sources que www.leroutier.net/Projects/
Nous utiliserons ici les sources suivantes :
`2003-02-12-prelude-perl-web-frontend.tar.gz`
On les décompresse dans le répertoir `/var/www/frontend-perl`.

Dans le fichier de configuration d'Apache, il y a entre autres :

```
User www-data
Group www-data
```

Ils définissent le profil utilisateur d'Apache. On va changer les droits des fichiers du frontend-perl selon ces paramètres :

```
chown -R www-data.www-data /var/www/frontend-perl
chmod -R u+x /var/www/frontend-perl/
```

D.8.3 Configuration d'Apache

Modifez le fichier `/etc/apache-ssl/httpd.conf`, pour avoir les lignes suivantes : `Listen 192.168.0.2`

```
DirectoryIndex index.pl index.html index.htm
ServerName localhost
DocumentRoot /var/www/
```

et rajouter les lignes suivantes :

```
     <Directory "/var/www/frontend-perl/">
       Options ExecCGI
       AddHandler cgi-script .pl
     </Directory>
```

Puis transformer

```
# If the perl module is installed, this will be enabled.
<IfModule mod_perl.c>
  Alias /perl/ /var/www/perl/
  <Location /perl>
    SetHandler perl-script
    PerlHandler Apache::Registry
    Options +ExecCGI
  </Location>
</IfModule>
```

en ce qui suit

```
# If the perl module is installed, this will be enabled.
<IfModule mod_perl.c>
  Alias /perl/ /var/www/frontend-perl/
  <Location /var/www/frontend-perl/>
    SetHandler perl-script
    PerlHandler Apache::Registry
    Options +ExecCGI
  </Location>
</IfModule>
```

D.8.4 Configuration de prelude-perl-frontend

Éditer le fichier `/var/www/frontend-perl/Functions/config.pl` pour y faire apparaître les lignes suivantes :

```
$conf{'dbtype'}='mysql';
$conf{'dbname'}='prelude';
$conf{'dbhost'}='localhost';
$conf{'dbport'}=3306; # default mysql port is 3306
$conf{'dblogin'}='prelude';
$conf{'dbpasswd'}='dessstri';
```

L'interface est maintenant accessible par *https ://localhost/frontend-perl/* avec le login *prelude* et le mot-de-passe *dessstri*.

D.9 Liens

Site officiel de Prelude-IDS : http ://www.prelude-ids.org

Dernière version de ce document : http ://lehmann.free.fr

Téléchargement des paquetages de Prelude-IDS :
http ://www.prelude-ids.org/rubrique.php3 ?id_rubrique=6

Autres documentations à propos de Prelude-IDS :
http ://www.prelude-ids.org/rubrique.php3 ?id_rubrique=1

Site officiel du frontend perl : http ://www.leroutier.net/Projects/

Site de Debian : http ://www.debian.org

Se procurer Debian sur CD : http ://ikarios.com/form/

.1 Liste de paquetages installés sur le système

Voici ce que donne l'exécution de la commande `dpkg -l` :

```
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Installed/Config-files/Unpacked/Failed-config/Half-installed
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: uppercase=ba
||/ Name           Version        Description
+++-==============-==============-============================================
ii  adduser        3.49           Add and remove users and groups
ii  apache-common  1.3.26-1.1     Support files for all Apache webservers
ii  apache-ssl     1.3.26.1+1.48- Versatile, high-performance HTTP server wit
ii  apt            0.5.4          Advanced front-end for dpkg
ii  apt-utils      0.5.4          APT utility programs
ii  at             3.1.8-11       Delayed job execution and batch processing
ii  aterm          0.4.2-4        Afterstep XVT - a VT102 emulator for the X
ii  base-config    1.54           Debian base configuration package
ii  base-files     3.0.7          Debian base system miscellaneous files
ii  base-passwd    3.4.2          Debian Base System Password/Group Files
ii  bash           2.05b-3        The GNU Bourne Again SHell
ii  bc             1.06-8         The GNU bc arbitrary precision calculator l
ii  biff           0.17.pre200004 a mail notification tool
ii  bin86          0.16.3-2       16-bit assembler and loader
ii  bind9-host     9.2.1-4        Version of 'host' bundled with BIND 9.X
ii  binutils       2.13.90.0.10-1 The GNU assembler, linker and binary utilit
ii  bison          1.75-1         A parser generator that is compatible with
ii  bsdmainutils   5.20020211-7   More utilities from FreeBSD.
ii  bsdutils       2.11n-4        Basic utilities from 4.4BSD-Lite.
ii  console-common 0.7.20         Basic infrastructure for text console confi
ii  console-data   2002.12.04dbs- Keymaps, fonts, charset maps, fallback tabl
ii  console-tools  0.2.3-23.3     Linux console and font utilities.
ii  console-tools- 0.2.3-23.3     Shared libraries for Linux console and font
ii  coreutils      4.5.2-1        The GNU core utilities
ii  cpio           2.5-1          GNU cpio -- a program to manage archives of
ii  cpp            2.95.4-17      The GNU C preprocessor.
ii  cpp-2.95       2.95.4-11woody The GNU C preprocessor.
ii  cpp-3.0        3.0.4-7        The GNU C preprocessor.
ii  cron           3.0pl1-72      management of regular background processing
ii  dc             1.06-8         The GNU dc arbitrary precision reverse-poli
ii  debconf        1.2.21         Debian configuration management system
ii  debianutils    1.16.7         Miscellaneous utilities specific to Debian
ii  defoma         0.11.1         Debian Font Manager -- automatic font confi
ii  dhcp-client    2.0pl5-14      DHCP Client
ii  dialog         0.9b-20020814- Displays user-friendly dialog boxes from sh
ii  diff           2.8.1-1        File comparison utilities
ii  dnsutils       9.2.1-4        Clients provided with BIND
ii  doc-debian     3.0.2          Debian Project documentation, Debian FAQ an
ii  doc-linux-text 2003.01-1      Linux HOWTOs, mini-HOWTOs, and FAQs in ASCI
ii  docbook        4.2-2          SGML DTD for authors of technical documenta
ii  docbook-dsssl  1.77-2         Modular DocBook DSSSL stylesheets, for prin
ii  docbook-to-man 2.0.0-10       Converter from DocBook SGML into roff -man
```

```
ii  dpkg           1.10.9          Package maintenance system for Debian
ii  dpkg-dev       1.10.9          Package building tools for Debian
ii  dselect        1.10.9          a user tool to manage Debian packages
ii  e2fsprogs      1.29+1.30-WIP-  The EXT2 file system utilities and librarie
ii  ed             0.2-19          The classic unix line editor
ii  emacs20        20.7-13.1       The GNU Emacs editor.
ii  emacsen-common 1.4.15          Common facilities for all emacsen.
ii  ethereal       0.9.5-2         Network traffic analyzer
ii  ethereal-commo 0.9.5-2         Network traffic analyser (common files)
ii  exim           3.36-3          An MTA (Mail Transport Agent)
ii  fdutils        5.4-20020222-3 Linux floppy utilities
ii  file           3.39-1          Determines file type using "magic" numbers
ii  fileutils      4.5.2-1         GNU file management utilities
ii  findutils      4.1.7-2         utilities for finding files--find, xargs, a
ii  finger         0.17-6          User information lookup program.
ii  flex           2.5.4a-27       A fast lexical analyzer generator.
ii  ftp            0.17-10         The FTP client.
ii  g++            2.95.4-17       The GNU C++ compiler.
ii  g++-2.95       2.95.4-11woody The GNU C++ compiler.
ii  gcc            2.95.4-17       The GNU C compiler.
ii  gcc-2.95       2.95.4-11woody The GNU C compiler.
ii  gcc-3.0        3.0.4-7         The GNU C compiler.
ii  gcc-3.0-base   3.0.4-7         The GNU Compiler Collection (base package).
ii  gdb            5.2.cvs2002040 The GNU Debugger
ii  gettext-base   0.10.40-8       GNU Internationalization utilities for the
ii  gnupg          1.2.0-1         GNU privacy guard - a free PGP replacement.
ii  gnupg-doc      2000.10.01-1    GNU privacy guard documentation.
ii  grep           2.4.2-3         GNU grep, egrep and fgrep.
ii  groff-base     1.18-7          GNU troff text-formatting system (base syst
ii  gsfonts        6.0-2.1         Fonts for the ghostscript interpreter
ii  gtk-doc-tools  1.0-4           GTK documentation tools
ii  gzip           1.3.5-1         The GNU compression utility.
ii  hermes1        1.3.2-3         The Hermes pixel-format library
ii  hostname       2.09            A utility to set/show the host name or doma
ii  iamerican      3.1.20.0-1      An American English dictionary for ispell.
ii  ibritish       3.1.20.0-1      A British English dictionary for ispell.
ii  ifupdown       0.6.4-4.4       High level tools to configure network inter
ii  info           4.2-1           Standalone GNU Info documentation browser
ii  ipchains       1.3.10-15       Network firewalling for Linux 2.2.x
ii  ipmasqadm      0.4.2-2         Utility for configuring extra masquerading
ii  iptables       1.2.7a-7        IP packet filter administration tools for 2
ii  ispell         3.1.20.0-1      International Ispell (an interactive spelli
ii  jade           1.2.1-28        James Clark’s DSSSL Engine
ii  kdelibs3       2.2.2-13        KDE core libraries (runtime files)
ii  kdelibs3-bin   2.2.2-13        KDE core binaries (binary files)
ii  kdm            2.2.2-14        The K Desktop Manager
ii  klogd          1.4.1-10        Kernel Logging Daemon
ii  less           378-2           A file pager program, similar to more(1)
ii  libbz2-1.0     1.0.2-1         A high-quality block-sorting file compresso
ii  libc6          2.2.5-14.3      GNU C Library: Shared libraries and Timezon
```

```
ii  libc6-dev      2.2.5-14.3     GNU C Library: Development Libraries and He
ii  libcap1        1.10-12        support for getting/setting POSIX.1e capabi
ii  libdate-calc-p 5.0-3          Perl library for accessing dates
ii  libdb1-compat  2.1.3-7        The Berkeley database routines [glibc 2.0/2
ii  libdb2         2.7.7.0-8      The Berkeley database routines (run-time fi
ii  libdb3         3.2.9-17       Berkeley v3 Database Libraries [runtime]
ii  libdb4.0       4.0.14-1       Berkeley v4.0 Database Libraries [runtime]
ii  libdbi-perl    1.21-2         The Perl5 Database Interface by Tim Bunce
ii  libdns5        9.2.1-4        DNS Shared Library used by BIND
ii  libdps1        4.2.1-3        Display PostScript (DPS) client library
ii  libexpat1      1.95.2-9       XML parsing C library - runtime library
ii  libfam0        2.6.8-3        client library to control the FAM daemon
ii  libfreetype6   2.1.2-9        FreeType 2 font engine, shared library file
ii  libgcc1        3.2.1-0pre3    GCC support library.
ii  libgd-graph-pe 1.35-3         Graph Plotting Module for Perl 5
ii  libgd-perl     1.40-1         Perl module wrapper for libgd
ii  libgd-text-per 0.83-3         Text utilities for use with GD
ii  libgd1         1.8.4-17       GD Graphics Library
ii  libgdbmg1      1.7.3-27.1     GNU dbm database routines (runtime version)
ii  libglib1.2     1.2.10-6       The GLib library of C routines
ii  libgtk1.2      1.2.10-14      The GIMP Toolkit set of widgets for X
ii  libgtk1.2-comm 1.2.10-14      Common files for the GTK+ library
ii  libident       0.22-2         simple RFC1413 client library - runtime
ii  libisc4        9.2.1-4        ISC Shared Library used by BIND
ii  libjpeg62      6b-6           The Independent JPEG Group's JPEG runtime l
ii  liblcms        1.08-3         Color management library
ii  libldap2       2.0.23-14      OpenLDAP libraries (without TLS support).
ii  liblockfile1   1.03           NFS-safe locking library, includes dotlockf
ii  liblwres1      9.2.1-4        Lightweight Resolver Library used by BIND
ii  libmm11        1.1.3-6        Shared memory library
ii  libmng1        1.0.3-3        Multiple-image Network Graphics library
ii  libmysqlclient 3.23.52-2      mysql database client library
ii  libmysqlclient 3.23.52-2      mysql database development files
ii  libncurses5    5.2.20020112a- Shared libraries for terminal handling
ii  libnewt0       0.50.17-9.6    Not Erik's Windowing Toolkit - text mode wi
ii  libnspr4       1.0.0-0.woody. Netscape Portable Runtime Library
ii  libnss-db      2.2-6          DB Name Service Module
ii  libnss3        1.0.0-0.woody. Network Security Service Libraries - runtim
ii  libpam-modules 0.76-7         Pluggable Authentication Modules for PAM
ii  libpam-runtime 0.76-7         Runtime support for the PAM library
ii  libpam0g       0.76-7         Pluggable Authentication Modules library
ii  libpaperg      1.1.8          Library for handling paper characteristics
ii  libpcap0       0.6.2-2        System interface for user-level packet capt
ii  libpcre3       3.4-1.1        Philip Hazel's Perl Compatible Regular Expr
ii  libperl5.6     5.6.1-8.2      Shared Perl library.
ii  libpng2        1.0.12-6       PNG library - runtime
ii  libpopt0       1.6.4-2        lib for parsing cmdline parameters
ii  libqt2         2.3.1-22       Qt GUI Library (runtime version).
ii  libreadline4   4.3-4          GNU readline and history libraries, run-tim
ii  libsasl7       1.5.27-3.3     Authentication abstraction library.
```

```
ii  libsp1          1.3.4-1.2.1-28 Runtime library for James Clark's SP suite
ii  libssl-dev      0.9.6g-6       SSL development libraries, header files and
ii  libssl0.9.6     0.9.6g-6       SSL shared libraries
ii  libstdc++2.10-  2.95.4-11woody The GNU stdc++ library (development files)
ii  libstdc++2.10-  2.95.4-11woody The GNU stdc++ library
ii  libstdc++3      3.0.4-7        The GNU stdc++ library version 3
ii  libtiff3g       3.5.5-6        Tag Image File Format library
ii  libungif4g      4.1.0b1-3      shared library for GIF images (runtime lib)
ii  libwrap0        7.6-ipv6.1-3   Wietse Venema's TCP wrappers library
ii  libwraster2     0.80.1-3       Shared libraries of Window Maker rasterizer
ii  libxaw7         4.2.1-3        X Athena widget set library
ii  libxml2         2.4.24-1       GNOME XML library
ii  libxml2-dev     2.4.24-1       Development files for the GNOME XML library
ii  libxslt1        1.0.21-0.2     XSLT processing library
ii  lilo            22.3.3-2       LInux LOader - The Classic OS loader can lo
ii  locales         2.2.5-14.3     GNU C Library: National Language (locale) d
ii  login           20000902-12    System login tools
ii  logrotate       3.6.5-2        Log rotation utility
ii  lpr             2000.05.07-4.2 BSD lpr/lpd line printer spooling system
ii  lsof            4.64-1         List open files.
ii  lynx            2.8.4.1b-3.2   Text-mode WWW Browser
ii  m4              1.4-14         a macro processing language
ii  mailx           8.1.2-0.200204 A simple mail user agent.
ii  make            3.79.1-15      The GNU version of the "make" utility.
ii  makedev         2.3.1-62       Creates device files in /dev.
ii  man-db          2.4.0-10       The on-line manual pager
ii  manpages        1.48-2         Manual pages about using a GNU/Linux system
ii  manpages-dev    1.48-2         Manual pages about using GNU/Linux for deve
ii  mawk            1.3.3-9        a pattern scanning and text processing lang
ii  mbr             1.1.5-1        Master Boot Record for IBM-PC compatible co
ii  mime-support    3.20-1         MIME files 'mime.types' & 'mailcap', and su
ii  modconf         0.2.44         Device Driver Configuration
ii  modutils        2.4.19-3       Linux module utilities.
ii  mount           2.11n-4        Tools for mounting and manipulating filesys
ii  mozilla         1.0.0-0.woody. Mozilla Web Browser - dummy package
ii  mozilla-browse  1.0.0-0.woody. Mozilla Web Browser - core and browser
ii  mozilla-mailne  1.0.0-0.woody. Mozilla Web Browser - mail and news support
ii  mozilla-psm     1.0.0-0.woody. Mozilla Web Browser - Personal Security Man
ii  mpack           1.5-9          Tools for encoding/decoding MIME messages.
ii  mtools          3.9.8-10       Tools for manipulating MSDOS files
ii  mtr-tiny        0.51-1         Full screen ncurses traceroute tool
ii  mutt            1.4.0-4        Text-based mailreader supporting MIME, GPG,
ii  mysql-client    3.23.52-2      mysql database client binaries
ii  mysql-common    3.23.52-2      mysql database common files (e.g. /etc/mysq
ii  mysql-server    3.23.52-2      mysql database server binaries
ii  nano            1.1.11-1       free Pico clone with some new features
ii  ncurses-base    5.2.20020112a- Descriptions of common terminal types
ii  ncurses-bin     5.2.20020112a- Terminal-related programs and man pages
ii  ncurses-term    5.2.20020112a- Additional terminal type definitions
ii  net-tools       1.60-4         The NET-3 networking toolkit
```

```
ii  netbase        4.09           Basic TCP/IP networking system
ii  netkit-inetd   0.10-9         The Internet Superserver
ii  netkit-ping    0.10-9         The ping utility from netkit
ii  nfs-common     1.0.2-1        NFS support files common to client and serv
ii  nfs-kernel-ser 1.0.2-1        Kernel NFS server support
ii  nmap           3.00-0.1       The Network Mapper
ii  nvi            1.79-20        4.4BSD re-implementation of vi.
ii  openssl        0.9.6g-6       Secure Socket Layer (SSL) binary and relate
ii  passwd         20000902-12    Change and administer password and group da
ii  patch          2.5.4-11       Apply a diff file to an original
ii  pciutils       2.1.10-3       Linux PCI Utilities (for 2.[12345].x kernel
ii  perl           5.6.1-8.2      Larry Wall's Practical Extraction and Repor
ii  perl-base      5.6.1-8.2      The Pathologically Eclectic Rubbish Lister.
ii  perl-modules   5.6.1-8.2      Core Perl modules.
ii  php4           4.1.2-4        A server-side, HTML-embedded scripting lang
ii  php4-mysql     4.1.2-4        MySQL module for php4
ii  pidentd        3.0.12-4       TCP/IP IDENT protocol server.
ii  portmap        5-2            The RPC portmapper
ii  ppp            2.4.1.uus-4    Point-to-Point Protocol (PPP) daemon.
ii  pppconfig      2.1            A text menu based utility for configuring p
ii  pppoe          3.3-1.2        PPP over Ethernet driver
ii  pppoeconf      0.9.10.8       configures PPPoE/ADSL
ii  procmail       3.22-4         Versatile e-mail processor.
ii  procps         2.0.7-10       The /proc file system utilities.
ii  psmisc         21.2-1         Utilities that use the proc filesystem
ii  python         2.1.3-4        An interactive object-oriented scripting la
ii  python-newt    0.50.17-9.6    A newt module for Python.
ii  python2.1      2.1.3-4        An interactive object-oriented scripting la
ii  rcs            5.7-13         The GNU Revision Control System
ii  reportbug      1.50           Reports bugs in the Debian distribution.
ii  sed            3.02-8         The GNU sed stream editor.
ii  setserial      2.17-30        Controls configuration of serial ports.
ii  sgml-base      1.17           utilities to maintain SGML catalog files
ii  sgml-data      1.8            Common SGML and XML DTDs and entities
ii  sharutils      4.2.1-9        shar, unshar, uuencode, uudecode
ii  shellutils     4.5.2-1        The GNU shell programming utilities.
ii  slang1         1.4.5-1        The S-Lang programming library - runtime ve
ii  sp             1.3.4-1.2.1-28 James Clark's SGML parsing tools
ii  ssh            3.4p1-4        Secure rlogin/rsh/rcp replacement (OpenSSH)
ii  strace         4.4-1.2        A system call tracer.
ii  sysklogd       1.4.1-10       System Logging Daemon
ii  syslinux       1.75-1         Bootloader for Linux/i386 using MS-DOS flop
ii  sysvinit       2.84-3         System-V like init.
ii  t1lib1         1.3.1-1        Type 1 font rasterizer library - runtime
ii  tar            1.13.25-3      GNU tar
ii  tasksel        1.21           Tool for selecting tasks for installation o
ii  tcpd           7.6-ipv6.1-3   Wietse Venema's TCP wrapper utilities
ii  tcsh           6.11.00-2.2    TENEX C Shell, an enhanced version of Berke
ii  telnet         0.17-19        The telnet client.
ii  texinfo        4.2-1          Documentation system for on-line informatio
```

```
ii  textutils      4.5.2-1        The GNU text file processing utilities
ii  time           1.7-15         The GNU time command.
ii  util-linux     2.11n-4        Miscellaneous system utilities.
ii  util-linux-loc 2.11n-4        Locales files for util-linux
ii  vacation       3.2.5          email autoresponder
ii  wenglish       2.0-2          English dictionary words for /usr/share/dic
ii  whiptail       0.50.17-9.6    Displays user-friendly dialog boxes from sh
ii  whois          4.5.31         The GNU whois client
ii  wmaker         0.80.1-3       NeXTSTEP-like window manager for X
ii  xbase-clients  4.2.1-3        miscellaneous X clients
ii  xfonts-100dpi  4.2.1-3        100 dpi fonts for X
ii  xfonts-base    4.2.1-3        standard fonts for X
ii  xfree86-common 4.2.1-3        X Window System (XFree86) infrastructure
ii  xfs            4.2.1-3        X font server
ii  xfs-xtt        1.3.0.xf420-8  X-TrueType font server
ii  xlibmesa3      4.2.1-3        XFree86 version of Mesa 3D graphics library
ii  xlibs          4.2.1-3        X Window System client libraries
ii  xpdf           1.01-3         Portable Document Format (PDF) suite
ii  xpdf-common    1.01-3         Portable Document Format (PDF) suite -- com
ii  xpdf-reader    1.01-3         Portable Document Format (PDF) suite -- vie
ii  xpdf-utils     1.01-3         Portable Document Format (PDF) suite -- uti
ii  xserver-common 4.2.1-3        files and utilities common to all X servers
ii  xserver-xfree8 4.2.1-3        the XFree86 X server
ii  xsltproc       1.0.21-0.2     XSLT command line processor
ii  xterm          4.2.1-3        X terminal emulator
ii  xutils         4.2.1-3        X Window System utility programs
ii  zlib1g         1.1.4-6        compression library - runtime
ii  zlib1g-dev     1.1.4-6        compression library - development
```

.2 Fichier httpd.conf

```
##
## httpd.conf -- Apache HTTP server configuration file
##

#
# Based upon the NCSA server configuration files originally by Rob McCool.
#
# This is the main Apache server configuration file.  It contains the
# configuration directives that give the server its instructions.
# See <URL:http://www.apache.org/docs/> for detailed information about
# the directives.
#
# Do NOT simply read the instructions in here without understanding
# what they do.  They're here only as hints or reminders.  If you are unsure
# consult the online docs. You have been warned.
#
# After this file is processed, the server will look for and process
```

```
# /etc/apache-ssl/srm.conf and then /etc/apache-ssl/access.conf
# unless you have overridden these with ResourceConfig and/or
# AccessConfig directives here.
#
# The configuration directives are grouped into three basic sections:
#  1. Directives that control the operation of the Apache server process as a
#     whole (the 'global environment').
#  2. Directives that define the parameters of the 'main' or 'default' server
#     which responds to requests that aren't handled by a virtual host.
#     These directives also provide default values for the settings
#     of all virtual hosts.
#  3. Settings for virtual hosts, which allow Web requests to be sent to
#     different IP addresses or hostnames and have them handled by the
#     same Apache server process.
#
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path.  If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so "logs/foo.log"
# with ServerRoot set to "/usr/local/apache" will be interpreted by the
# server as "/usr/local/apache/logs/foo.log".
#

### Section 1: Global Environment
#
# The directives in this section affect the overall operation of Apache,
# such as the number of concurrent requests it can handle or where it
# can find its configuration files.
#

#
# ServerType is either inetd, or standalone.  Inetd mode is only supported on
# Unix platforms.
# SSL Servers MUST be standalone, currently.
#
ServerType standalone

#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept, unless they are specified
# with an absolute path.
#
# NOTE!  If you intend to place this on an NFS (or otherwise network)
# mounted filesystem then please read the LockFile documentation
# (available at <URL:http://www.apache.org/docs/mod/core.html#lockfile>);
# you will save yourself a lot of trouble.
#
# Do NOT add a slash at the end of the directory path.
#
ServerRoot /etc/apache-ssl
```

```
#
# The LockFile directive sets the path to the lockfile used when Apache
# is compiled with either USE_FCNTL_SERIALIZED_ACCEPT or
# USE_FLOCK_SERIALIZED_ACCEPT. This directive should normally be left at
# its default value. The main reason for changing it is if the logs
# directory is NFS mounted, since the lockfile MUST BE STORED ON A LOCAL
# DISK. The PID of the main server process is automatically appended to
# the filename.
#
LockFile /var/lock/apache.lock

#
# PidFile: The file in which the server should record its process
# identification number when it starts.
#
PidFile /var/run/apache-ssl.pid

#
# ScoreBoardFile: File used to store internal server process information.
# Not all architectures require this.  But if yours does (you'll know because
# this file will be  created when you run Apache) then you *must* ensure that
# no two invocations of Apache share the same scoreboard file.
#
ScoreBoardFile /var/run/apache-ssl.scoreboard

#
# In the standard configuration, the server will process this file,
# srm.conf, and access.conf in that order.  The latter two files are
# now distributed empty, as it is recommended that all directives
# be kept in a single file for simplicity.  The commented-out values
# below are the built-in defaults.  You can have the server ignore
# these files altogether by using "/dev/null" (for Unix) or
# "nul" (for Win32) for the arguments to the directives.
#
#ResourceConfig /etc/apache-ssl/srm.conf
#AccessConfig /etc/apache-ssl/access.conf

#
# Timeout: The number of seconds before receives and sends time out.
#
Timeout 300

#
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
#
KeepAlive On
```

```
#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
#
MaxKeepAliveRequests 100

#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout 15

#
# Server-pool size regulation.  Rather than making you guess how many
# server processes you need, Apache dynamically adapts to the load it
# sees --- that is, it tries to maintain enough server processes to
# handle the current load, plus a few spare servers to handle transient
# load spikes (e.g., multiple simultaneous requests from a single
# Netscape browser).
#
# It does this by periodically checking how many servers are waiting
# for a request.  If there are fewer than MinSpareServers, it creates
# a new spare.  If there are more than MaxSpareServers, some of the
# spares die off.  The default values are probably OK for most sites.
#
MinSpareServers 5
MaxSpareServers 10

#
# Number of servers to start initially --- should be a reasonable ballpark
# figure.
#
StartServers 5

#
# Limit on total number of servers running, i.e., limit on the number
# of clients who can simultaneously connect --- if this limit is ever
# reached, clients will be LOCKED OUT, so it should NOT BE SET TOO LOW.
# It is intended mainly as a brake to keep a runaway server from taking
# the system with it as it spirals down...
#
MaxClients 150

#
# MaxRequestsPerChild: the number of requests each child process is
# allowed to process before the child dies.  The child will exit so
# as to avoid problems after prolonged use when Apache (and maybe the
# libraries it uses) leak memory or other resources.  On most systems, this
# isn’t really needed, but a few (such as Solaris) do have notable leaks
```

```
# in the libraries. For these platforms, set to something like 10000
# or so; a setting of 0 means unlimited.
#
# NOTE: This value does not include keepalive requests after the initial
#       request per connection. For example, if a child process handles
#       an initial request and 10 subsequent "keptalive" requests, it
#       would only count as 1 request towards this limit.
#
MaxRequestsPerChild 100

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, in addition to the default. See also the <VirtualHost>
# directive.
#
#Listen 3000
Listen 192.168.0.2

#
# BindAddress: You can support virtual hosts with this option. This directive
# is used to tell the server which IP address to listen to. It can either
# contain "*", an IP address, or a fully qualified Internet domain name.
# See also the <VirtualHost> and Listen directives.
#
#BindAddress *

#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO yo
# have to place corresponding ‘LoadModule’ lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Please read the file README.DSO in the Apache 1.3 distribution for more
# details about the DSO mechanism and run ‘apache -l’ for the list of already
# built-in (statically linked and thus always available) modules in your apac
# binary.
#
# Please keep this LoadModule: line here, it is needed for installation.
# LoadModule vhost_alias_module /usr/lib/apache/1.3/mod_vhost_alias.so
# LoadModule env_module /usr/lib/apache/1.3/mod_env.so
LoadModule config_log_module /usr/lib/apache/1.3/mod_log_config_ssl.so
LoadModule mime_magic_module /usr/lib/apache/1.3/mod_mime_magic.so
LoadModule mime_module /usr/lib/apache/1.3/mod_mime_ssl.so
LoadModule negotiation_module /usr/lib/apache/1.3/mod_negotiation.so
LoadModule status_module /usr/lib/apache/1.3/mod_status.so
# LoadModule info_module /usr/lib/apache/1.3/mod_info.so
# LoadModule includes_module /usr/lib/apache/1.3/mod_include.so
LoadModule autoindex_module /usr/lib/apache/1.3/mod_autoindex.so
LoadModule dir_module /usr/lib/apache/1.3/mod_dir.so
LoadModule cgi_module /usr/lib/apache/1.3/mod_cgi.so
```

```
# LoadModule asis_module /usr/lib/apache/1.3/mod_asis.so
# LoadModule imap_module /usr/lib/apache/1.3/mod_imap.so
# LoadModule action_module /usr/lib/apache/1.3/mod_actions.so
# LoadModule speling_module /usr/lib/apache/1.3/mod_speling.so
LoadModule userdir_module /usr/lib/apache/1.3/mod_userdir.so
LoadModule alias_module /usr/lib/apache/1.3/mod_alias.so
LoadModule rewrite_module /usr/lib/apache/1.3/mod_rewrite.so
LoadModule access_module /usr/lib/apache/1.3/mod_access.so
LoadModule auth_module /usr/lib/apache/1.3/mod_auth_ssl.so
# LoadModule anon_auth_module /usr/lib/apache/1.3/mod_auth_anon.so
# LoadModule dbm_auth_module /usr/lib/apache/1.3/mod_auth_dbm.so
# LoadModule db_auth_module /usr/lib/apache/1.3/mod_auth_db.so
# LoadModule proxy_module /usr/lib/apache/1.3/libproxy.so
# LoadModule digest_module /usr/lib/apache/1.3/mod_digest.so
# LoadModule cern_meta_module /usr/lib/apache/1.3/mod_cern_meta.so
LoadModule expires_module /usr/lib/apache/1.3/mod_expires.so
# LoadModule headers_module /usr/lib/apache/1.3/mod_headers.so
# LoadModule usertrack_module /usr/lib/apache/1.3/mod_usertrack.so
LoadModule unique_id_module /usr/lib/apache/1.3/mod_unique_id.so
LoadModule setenvif_module /usr/lib/apache/1.3/mod_setenvif.so
# LoadModule sys_auth_module /usr/lib/apache/1.3/mod_auth_sys.so
# LoadModule put_module /usr/lib/apache/1.3/mod_put.so
# LoadModule throttle_module /usr/lib/apache/1.3/mod_throttle.so
LoadModule apache_ssl_module /usr/lib/apache/1.3/libssl.so
# LoadModule allowdev_module /usr/lib/apache/1.3/mod_allowdev.so
# LoadModule eaccess_module /usr/lib/apache/1.3/mod_eaccess.so
# LoadModule roaming_module /usr/lib/apache/1.3/mod_roaming.so
LoadModule php4_module /usr/lib/apache/1.3/libphp4.so


#
# ExtendedStatus: controls whether Apache will generate "full" status
# information (ExtendedStatus On) or just basic information (ExtendedStatus
# Off) when the "server-status" handler is called. The default is Off.
#
ExtendedStatus On

### Section 2: 'Main' server configuration
#
# The directives in this section set up the values used by the 'main'
# server, which responds to any requests that aren't handled by a
# <VirtualHost> definition.  These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.
#
# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
#

#
```

```
# If your ServerType directive (set earlier in the 'Global Environment'
# section) is set to "inetd", the next few directives don't have any
# effect since their settings are defined by the inetd configuration.
# Skip ahead to the ServerAdmin directive.
#

#
# Port: The port to which the standalone server listens. For
# ports < 1023, you will need apache to be run as root initially.
#
# The default port for SSL is 443...

Port 443

#
# If you wish apache to run as a different user or group, you must run
# apacheas root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run apache as.
#  . On SCO (ODT 3) use "User nouser" and "Group nogroup".
#  . On HPUX you may not be able to use shared memory as nobody, and the
#    suggested workaround is to create a user www and use that user.
#  NOTE that some kernels refuse to setgid(Group) or semctl(IPC_SET)
#  when the value of (unsigned)Group is above 60000;
#  don't use Group nobody on these systems!
#
User www-data
Group www-data

#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed.  This address appears on some server-generated pages, such
# as error documents.
#
ServerAdmin webmaster@manager

#
# ServerName: allows you to set a host name which is sent back to clients for
# your server if it's different than the one the program would get (i.e., use
# "www" instead of the host's real name).
#
# Note: You cannot just invent host names and hope they work. The name you
# define here must be a valid DNS name for your host. If you don't understand
# this, ask your network administrator.
# If your host doesn't have a registered DNS name, enter its IP address here.
# You will have to access it by its address (e.g., http://123.45.67.89/)
# anyway, and this will make redirections work in a sensible way.
#
ServerName localhost
```

```
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot /var/www

#
# Each directory to which Apache has access, can be configured with respect
# to which services and features are allowed and/or disabled in that
# directory (and its subdirectories).
#
# First, we configure the "default" to be a very restrictive set of
# permissions.
#
<Directory />
    Options SymLinksIfOwnerMatch
    AllowOverride None
</Directory>

#
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
#

#
# This should be changed to whatever you set DocumentRoot to.
#
<Directory /var/www/>

#
# This may also be "None", "All", or any combination of "Indexes",
# "Includes", "FollowSymLinks", "ExecCGI", or "MultiViews".
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
    Options Indexes Includes FollowSymLinks MultiViews

#
# This controls which options the .htaccess files in directories can
# override. Can also be "All", or any combination of "Options", "FileInfo",
# "AuthConfig", and "Limit"
#
    AllowOverride None

#
# Controls who can get stuff from this server.
```

```
#
    Order allow,deny
    Allow from all
</Directory>

#
# UserDir: The name of the directory which is appended onto a user's home
# directory if a ~user request is received.
#
<IfModule mod_userdir.c>
    UserDir public_html
</IfModule>

#
# Control access to UserDir directories.  The following is an example
# for a site where these directories are restricted to read-only.
#
<Directory /home/*/public_html>
    AllowOverride FileInfo AuthConfig Limit
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    <Limit GET POST OPTIONS PROPFIND>
        Order allow,deny
        Allow from all
    </Limit>
    <Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
        Order deny,allow
        Deny from all
    </Limit>
</Directory>

#
# DirectoryIndex: Name of the file or files to use as a pre-written HTML
# directory index.  Separate multiple entries with spaces.
#
<IfModule mod_dir.c>
    DirectoryIndex index.pl index.php index.html index.htm index.shtml index.
</IfModule>

#
# AccessFileName: The name of the file to look for in each directory
# for access control information.
#
AccessFileName .htaccess

#
# The following lines prevent .htaccess files from being viewed by
# Web clients.  Since .htaccess files often contain authorization
# information, access is disallowed for security reasons.  Comment
# these lines out if you want Web visitors to see the contents of
# .htaccess files.  If you change the AccessFileName directive above,
```

```
# be sure to make the corresponding changes here.
#
# Also, folks tend to use names such as .htpasswd for password
# files, so this will protect those as well.
#
<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
</Files>

#
# CacheNegotiatedDocs: By default, Apache sends "Pragma: no-cache" with each
# document that was negotiated on the basis of content. This asks proxy
# servers not to cache the document. Uncommenting the following line disables
# this behavior, and proxies will be allowed to cache the documents.
#
#CacheNegotiatedDocs

#
# UseCanonicalName:  (new for 1.3)  With this setting turned on, whenever
# Apache needs to construct a self-referencing URL (a URL that refers back
# to the server the response is coming from) it will use ServerName and
# Port to form a "canonical" name.  With this setting off, Apache will
# use the hostname:port that the client supplied, when possible.  This
# also affects SERVER_NAME and SERVER_PORT in CGI scripts.
#
UseCanonicalName On

#
# TypesConfig describes where the mime.types file (or equivalent) is
# to be found.
#
TypesConfig /etc/mime.types

#
# DefaultType is the default MIME type the server will use for a document
# if it cannot otherwise determine one, such as from filename extensions.
# If your server contains mostly text or HTML documents, "text/plain" is
# a good value.  If most of your content is binary, such as applications
# or images, you may want to use "application/octet-stream" instead to
# keep browsers from trying to display binary files as though they are
# text.
#
DefaultType text/plain

#
# The mod_mime_magic module allows the server to use various hints from the
# contents of the file itself to determine its type.  The MIMEMagicFile
# directive tells the module where the hint definitions are located.
# mod_mime_magic is not part of the default server (you have to add
```

```
# it yourself with a LoadModule [see the DSO paragraph in the 'Global
# Environment' section], or recompile the server and include mod_mime_magic
# as part of the configuration), so it's enclosed in an <IfModule> container.
# This means that the MIMEMagicFile directive will only be processed if the
# module is part of the server.
#
<IfModule mod_mime_magic.c>
    MIMEMagicFile share/magic
</IfModule>

#
# HostnameLookups: Log the names of clients or just their IP addresses
# e.g., www.apache.org (on) or 204.62.129.132 (off).
# The default is off because it'd be overall better for the net if people
# had to knowingly turn this feature on, since enabling it means that
# each client request will result in AT LEAST one lookup request to the
# nameserver.
#
HostnameLookups Off

# Note that Log files are now rotated by logrotate, not by apache itself.
# This means that apache no longer attempts to magically determine
# where your log files are kept; you have to fill out stanzas in
# /etc/logrotate.d/apache-ssl yourself.

#
# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here.  If you *do* define an error logfile for a <VirtualHost>
# container, that host's errors will be logged there and not here.
#
ErrorLog /var/log/apache-ssl/error.log

#
# LogLevel: Control the number of messages logged to the error_log.
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
#
LogLevel warn

#
# The following directives define some format nicknames for use with
# a CustomLog directive (see below).
#
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %T %v
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %P %T
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" comb
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
```

```
LogFormat "%{User-agent}i" agent

#
# The location and format of the access logfile (Common Logfile Format).
# If you do not define any access logfiles within a <VirtualHost>
# container, they will be logged here.  Contrariwise, if you *do*
# define per-<VirtualHost> access logfiles, transactions will be
# logged therein and *not* in this file.
#
#CustomLog /var/log/apache-ssl/access.log common

#
# If you would like to have agent and referer logfiles, uncomment the
# following directives.
#
#CustomLog /var/log/apache-ssl/referer.log referer
#CustomLog /var/log/apache-ssl/agent.log agent

#
# If you prefer a single logfile with access, agent, and referer information
# (Combined Logfile Format) you can use the following directive.
#
CustomLog /var/log/apache-ssl/access.log combined

#
# Optionally add a line containing the server version and virtual host
# name to server-generated pages (error documents, FTP directory listings,
# mod_status and mod_info output etc., but not CGI generated documents).
# Set to "EMail" to also include a mailto: link to the ServerAdmin.
# Set to one of:  On | Off | EMail
#
ServerSignature On

#
# Aliases: Add here as many aliases as you need (with no limit). The format i
# Alias fakename realname
#
# Note that if you include a trailing / on fakename then the server will
# require it to be present in the URL.  So "/icons" isn't aliased in this
# example, only "/icons/"..
#

Alias /icons/ /usr/share/apache/icons/

<Directory /usr/share/apache/icons>
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

```
#
# ScriptAlias: This controls which directories contain server scripts.
# ScriptAliases are essentially the same as Aliases, except that
# documents in the realname directory are treated as applications and
# run by the server when requested rather than as documents sent to the clien
# The same rules about trailing "/" apply to ScriptAlias directives as to
# Alias.
#
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/

#
# "/usr/lib/cgi-bin" could be changed to whatever your ScriptAliased
# CGI directory exists, if you have that configured.
#
<Directory /usr/lib/cgi-bin/>
    AllowOverride None
    Options ExecCGI
    Order allow,deny
    Allow from all
</Directory>

#
# Redirect allows you to tell clients about documents which used to exist in
# your server's namespace, but do not anymore. This allows you to tell the
# clients where to look for the relocated document.
# Format: Redirect old-URI new-URL
#

#
# Directives controlling the display of server-generated directory listings.
#

<IfModule mod_autoindex.c>

    #
    # FancyIndexing: whether you want fancy directory indexing or standard
    #
    IndexOptions FancyIndexing NameWidth=*

    #
    # AddIcon* directives tell the server which icon to show for different
    # files or filename extensions.  These are only displayed for
    # FancyIndexed directories.
    #
    AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip

    AddIconByType (TXT,/icons/text.gif) text/*
    AddIconByType (IMG,/icons/image2.gif) image/*
    AddIconByType (SND,/icons/sound2.gif) audio/*
```

```
AddIconByType (VID,/icons/movie.gif) video/*

AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core
AddIcon /icons/deb.gif .deb

AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^

#
# DefaultIcon: which icon to show for files which do not have an icon
# explicitly set.
#
DefaultIcon /icons/unknown.gif

#
# AddDescription: allows you to place a short description after a file in
# server-generated indexes.  These are only displayed for FancyIndexed
# directories.
# Format: AddDescription "description" filename
#
#AddDescription "GZIP compressed document" .gz
#AddDescription "tar archive" .tar
#AddDescription "GZIP compressed tar archive" .tgz

#
# ReadmeName: the name of the README file the server will look for by
# default, and append to directory listings.
#
# HeaderName: the name of a file which should be prepended to
# directory indexes.
#
# The server will first look for name.html and include it if found.
# If name.html doesn’t exist, the server will then look for name.txt
```

```
    # and include it as plaintext if found.
    #
    ReadmeName README
    HeaderName HEADER

    #
    # IndexIgnore: a set of filenames which directory indexing should ignore
    # and not include in the listing.  Shell-style wildcarding is permitted.
    #
    IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t

</IfModule>

#
# Document types.
#
<IfModule mod_mime.c>

    # AddEncoding allows you to have certain browsers (Mosaic/X 2.1+)
    # uncompress information on the fly. Note: Not all browsers support
    # this.  Despite the name similarity, the following Add* directives
    # have nothing to do with the FancyIndexing customization
    # directives above.

    AddEncoding x-compress Z
    AddEncoding x-gzip gz tgz

    #
    # AddLanguage: allows you to specify the language of a document. You can
    # then use content negotiation to give a browser a file in a language
    # it can understand.
    #
    # Note 1: The suffix does not have to be the same as the language
    # keyword --- those with documents in Polish (whose net-standard
    # language code is pl) may wish to use "AddLanguage pl .po" to
    # avoid the ambiguity with the common suffix for perl scripts.
    #
    # Note 2: The example entries below illustrate that in quite
    # some cases the two character 'Language' abbriviation is not
    # identical to the two character 'Country' code for its country,
    # E.g. 'Danmark/dk' versus 'Danish/da'.
    #
    # Note 3: In the case of 'ltz' we violate the RFC by using a three char
    # specifier. But there is 'work in progress' to fix this and get
    # the reference data for rfc1766 cleaned up.
    #
    # Danish (da) - Dutch (nl) - English (en) - Estonian (ee)
    # French (fr) - German (de) - Greek-Modern (el)
    # Italian (it) - Portugese (pt) - Luxembourgeois* (ltz)
    # Spanish (es) - Swedish (sv) - Catalan (ca) - Czech(cz)
```

```
# Polish (pl) - Brazilian Portuguese (pt-br) - Japanese (ja)
#
AddLanguage da .dk
AddLanguage nl .nl
AddLanguage en .en
AddLanguage et .ee
AddLanguage fr .fr
AddLanguage de .de
AddLanguage el .el
AddLanguage it .it
AddLanguage ja .ja
AddCharset ISO-2022-JP .jis
AddLanguage pl .po
AddCharset ISO-8859-2 .iso-pl
AddLanguage pt .pt
AddLanguage pt-br .pt-br
AddLanguage ltz .lu
AddLanguage ca .ca
AddLanguage es .es
AddLanguage sv .se
AddLanguage cz .cz

# LanguagePriority: allows you to give precedence to some languages
# in case of a tie during content negotiation.
#
# Just list the languages in decreasing order of preference. We have
# more or less alphabetized them here. You probably want to change
# this.
#
<IfModule mod_negotiation.c>
    LanguagePriority en da nl et fr de el it ja pl pt pt-br ltz ca es sv
</IfModule>

#
# AddType allows you to tweak mime.types without actually editing
# it, or to make certain files to be certain types.
#
# For example, the PHP 3.x module (not part of the Apache
# distribution - see http://www.php.net) will typically use:
#
#AddType application/x-httpd-php3 .php3
#AddType application/x-httpd-php3-source .phps
#
# And for PHP 4.x, use:
#
#AddType application/x-httpd-php .php
#AddType application/x-httpd-php-source .phps

AddType application/x-tar .tgz
AddType image/bmp .bmp
```

```
    # hdml
    AddType text/x-hdml .hdml

    #
    # AddHandler allows you to map certain file extensions to "handlers",
    # actions unrelated to filetype. These can be either built into
    # the server or added with the Action command (see below).
    #
    # If you want to use server side includes, or CGI outside
    # ScriptAliased directories, uncomment the following lines.
    #
    # To use CGI scripts:
    #
    #AddHandler cgi-script .cgi .sh .pl

    #
    # To use server-parsed HTML files
    #
    #AddType text/html .shtml
    #AddHandler server-parsed .shtml

    #
    # Uncomment the following line to enable Apache's send-asis HTTP
    # file feature.
    #
    #AddHandler send-as-is asis

    #
    # If you wish to use server-parsed imagemap files, use
    #
    #AddHandler imap-file map

    #
    # To enable type maps, you might want to use
    #
    #AddHandler type-map var

</IfModule>
# End of document types.

# Default charset to iso-8859-1 (ttp://www.apache.org/info/css-security/).

AddDefaultCharset on

#
# Action: lets you define media types that will execute a script whenever
# a matching file is called. This eliminates the need for repeated URL
# pathnames for oft-used CGI file processors.
# Format: Action media/type /cgi-script/location
```

```
# Format: Action handler-name /cgi-script/location
#

#
# MetaDir: specifies the name of the directory in which Apache can find
# meta information files. These files contain additional HTTP headers
# to include when sending the document
#
#MetaDir .web

#
# MetaSuffix: specifies the file name suffix for the file containing the
# meta information.
#
#MetaSuffix .meta

#
# Customizable error response (Apache style)
#  these come in three flavors
#
#    1) plain text
#ErrorDocument 500 "The server made a boo boo.
#  n.b.  the (") marks it as text, it does not get output
#
#    2) local redirects
#ErrorDocument 404 /missing.html
#  to redirect to local URL /missing.html
#ErrorDocument 404 /cgi-bin/missing_handler.pl
#  N.B.: You can redirect to a script or a document using server-side-include
#
#    3) external redirects
#ErrorDocument 402 http://some.other_server.com/subscription_info.html
#  N.B.: Many of the environment variables associated with the original
#  request will *not* be available to such a script.

<IfModule mod_setenvif.c>
    #
    # The following directives modify normal HTTP response behavior.
    # The first directive disables keepalive for Netscape 2.x and browsers th
    # spoof it. There are known problems with these browser implementations.
    # The second directive is for Microsoft Internet Explorer 4.0b2
    # which has a broken HTTP/1.1 implementation and does not properly
    # support keepalive when it is used on 301 or 302 (redirect) responses.
    #
    BrowserMatch "Mozilla/2" nokeepalive
    BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0

    #
    # The following directive disables HTTP/1.1 responses to browsers which
    # are in violation of the HTTP/1.0 spec by not being able to grok a
```

```
    # basic 1.1 response.
    #
    BrowserMatch "RealPlayer 4\.0" force-response-1.0
    BrowserMatch "Java/1\.0" force-response-1.0
    BrowserMatch "JDK/1\.0" force-response-1.0
</IfModule>


# If the perl module is installed, this will be enabled.
<IfModule mod_perl.c>
  Alias /perl/ /var/www/frontend-perl/
  <Location /var/www/frontend-perl/>
    SetHandler perl-script
    PerlHandler Apache::Registry
    Options +ExecCGI
  </Location>
</IfModule>

# PerlModule Apache::DBI

#     <Files *.pl>
#       SetHandler perl-script
#       PerlHandler Apache::PerlRun
#       PerlSendHeader On
#     </Files>


     <Directory "/var/www/frontend-perl/">
       Options ExecCGI
       AddHandler cgi-script .pl
     </Directory>



#
# Allow http put (such as Netscape Gold's publish feature)
# Use htpasswd to generate /etc/apache/passwd.
# You must unremark these two lines at the top of this file as well:
#LoadModule put_module modules/mod_put.so
#
#Alias /upload /tmp
#<Location /upload>
#    EnablePut On
#    AuthType Basic
#    AuthName Temporary
#    AuthUserFile /etc/apache/passwd
#    EnableDelete Off
#    umask 007
#    <Limit PUT>
# require valid-user
```

```
#    </Limit>
#</Location>

#
# Allow server status reports, with the URL of http://servername/server-statu
# Change the ".your_domain.com" to match your domain to enable.
#
#<Location /server-status>
#    SetHandler server-status
#    Order deny,allow
#    Deny from all
#    Allow from .your_domain.com
#</Location>

#
# Allow remote server configuration reports, with the URL of
#  http://servername/server-info (requires that mod_info.c be loaded).
# Change the ".your_domain.com" to match your domain to enable.
#
#<Location /server-info>
#    SetHandler server-info
#    Order deny,allow
#    Deny from all
#    Allow from .your_domain.com
#</Location>

# Allow access to local system documentation from localhost.
# (Debian Policy assumes /usr/share/doc is "/doc/", at least from the localho
Alias /doc/ /usr/share/doc/

<Location /doc>
  order deny,allow
  deny from all
  allow from 127.0.0.0/255.0.0.0
  Options Indexes FollowSymLinks MultiViews
</Location>

#
# There have been reports of people trying to abuse an old bug from pre-1.1
# days.  This bug involved a CGI script distributed as a part of Apache.
# By uncommenting these lines you can redirect these attacks to a logging
# script on phf.apache.org.  Or, you can record them yourself, using the scri
# support/phf_abuse_log.cgi.
#
#<Location /cgi-bin/phf*>
#    Deny from all
#    ErrorDocument 403 http://phf.apache.org/phf_abuse_log.cgi
#</Location>

<IfModule mod_proxy.c>
```

```
    #
    # Proxy Server directives. Uncomment the following lines to
    # enable the proxy server:
    #
    #<IfModule mod_proxy.c>
    #ProxyRequests On
    #
    #<Directory proxy:*>
    #    Order deny,allow
    #    Deny from all
    #    Allow from .your_domain.com
    #</Directory>
</IfModule>

#
# Enable/disable the handling of HTTP/1.1 "Via:" headers.
# ("Full" adds the server version; "Block" removes all outgoing Via: headers)
# Set to one of: Off | On | Full | Block
#
#ProxyVia On

#
# To enable the cache as well, edit and uncomment the following lines:
# (no cacheing without CacheRoot)
#
#CacheRoot "/var/cache/apache"
#CacheSize 5
#CacheGcInterval 4
#CacheMaxExpire 24
#CacheLastModifiedFactor 0.1
#CacheDefaultExpire 1
#NoCache a_domain.com another_domain.edu joes.garage_sale.com

#</IfModule>
# End of proxy directives.

### Section 3: Virtual Hosts
#
# VirtualHost: If you want to maintain multiple domains/hostnames on your
# machine you can setup VirtualHost containers for them.
# Please see the documentation at <URL:http://www.apache.org/docs/vhosts/>
# for further details before you try to setup virtual hosts.
# You may use the command line option '-S' to verify your virtual host
# configuration.

#
# If you want to use name-based virtual hosts you need to define at
# least one IP address (and port number) for them.
#
#NameVirtualHost 12.34.56.78:80
```

```
#NameVirtualHost 12.34.56.78



#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
#
#<VirtualHost ip.address.of.host.some_domain.com>
#    ServerAdmin webmaster@host.some_domain.com
#    DocumentRoot /www/docs/host.some_domain.com
#    ServerName host.some_domain.com
#    ErrorLog logs/host.some_domain.com-error.log
#    CustomLog logs/host.some_domain.com-access.log common
#</VirtualHost>

#<VirtualHost _default_:*>
#</VirtualHost>

# ------------------------------SSL-----------------------------------
# This is an example configuration file for Apache-SSL.
# Copyright (C) 1995,6,7 Ben Laurie

# By popular demand, this file now illustrates the way to create two websites
# one secured (on port 8887), the other not (on port 8888).

# You may need one of thse
#User webuser
#User ben
#Group group

# SSL Servers MUST be standalone, currently.
#ServerType standalone

# The default port for SSL is 443...
#Port 8887
#Listen ServerPort
Listen 443

# My test document root
#DocumentRoot /u/ben/www/1/docs
#DocumentRoot /u/ben/apache/apache_1.3.0-ssl/htdocs

#<Directory /u/ben/apache/apache_1.3.0-ssl/htdocs/manual>
# This directive forbids access except when SSL is in use. Very handy for
# defending against configuration errors that expose stuff that should be
# protected
#SSLRequireSSL
#</Directory>
```

```
# Watch what’s going on
#TransferLog /var/log/apache-ssl/transfer.log

# Note that all SSL options can apply to virtual hosts.

# Disable SSL. Useful in combination with virtual hosts. Note that SSLEnable
# now also supported.
SSLEnable

# Set the path for the global cache server executable.
# If this facility gives you trouble, you can disable it by setting
# CACHE_SESSIONS to FALSE in apache_ssl.c
SSLCacheServerPath /usr/lib/apache-ssl/gcache

# Set the global cache server port number, or path. If it is a path, a Unix
# domain socket is used. If a number, a TCP socket.
SSLCacheServerPort /var/run/gcache_port
#SSLCacheServerPort 1234

# Set the session cache timeout, in seconds (set to 15 for testing, use a
# higher value in real life)
SSLSessionCacheTimeout 15

# Set the CA certificate verification path (must be PEM encoded).
# (in addition to getenv("SSL_CERT_DIR"), I think).
#SSLCACertificatePath /u/ben/apache/apache_1.2.5-ssl/SSLconf/conf
SSLCACertificatePath /etc/apache-ssl

# Set the CA certificate verification file (must be PEM encoded).
# (in addition to getenv("SSL_CERT_FILE"), I think).
#SSLCACertificateFile /some/where/somefile
#SSLCACertificateFile /u/ben/apache/apache_1.2.5-ssl/SSLconf/conf/httpsd.pem

# Point SSLCertificateFile at a PEM encoded certificate.
# If the certificate is encrypted, then you will be prompted for a pass phras
# Note that a kill -1 will prompt again.
# A test certificate can be generated with "make certificate".
SSLCertificateFile /etc/apache-ssl/apache.pem
#SSLCertificateFile /u/ben/apache/apache_1.2.6-ssl/SSLconf/conf/t1.pem


# If the key is not combined with the certificate, use this directive to
# point at the key file. If this starts with a ’/’ it specifies an absolute
# path, otherwise it is relative to the default certificate area. That is, it
# means "<default>/private/<keyfile>".
#SSLCertificateKeyFile /some/place/with/your.key

# Set SSLVerifyClient to:
# 0 if no certicate is required
```

```
# 1 if the client may present a valid certificate
# 2 if the client must present a valid certificate
# 3 if the client may present a valid certificate but it is not required to
#   have a valid CA
SSLVerifyClient 0
# How deeply to verify before deciding they don’t have a valid certificate
SSLVerifyDepth 10

# Translate the client X509 into a Basic authorisation. This means that the
# standard Auth/DBMAuth methods can be used for access control. The user name
# is the "one line" version of the client’s X509 certificate. Note that no
# password is obtained from the user. Every entry in the user file needs this
# password: xxj31ZMTZzkVA. See the code for further explanation.
SSLFakeBasicAuth

# List the ciphers that the client is permitted to negotiate. See the source
# for a definitive list. For example:
#SSLRequiredCiphers RC4-MD5:RC4-SHA:IDEA-CBC-MD5:DES-CBC3-SHA

# These two can be used per-directory to require or ban ciphers. Note that (a
# least in the current version) Apache-SSL will not attempt to renegotiate if
# cipher is banned (or not required).
#SSLRequireCipher
#SSLBanCipher

# A home for miscellaneous rubbish generated by SSL. Much of it is duplicated
# in the error log file. Put this somewhere where it cannot be used for symli
# attacks on a real server (i.e. somewhere where only root can write).
#SSLLogFile /var/log/ssl.log

# Custom logging
CustomLog /var/log/apache-ssl/ssl.log "%t %{version}c %{cipher}c %{clientcert

#<VirtualHost scuzzy:8888>
#SSLDisable
#SSLEnable
#</VirtualHost>

# If you want, you can disable SSL globally, and enable it in a virtual host.
#<VirtualHost scuzzy:8887>
#SSLEnable
# and the rest of the SSL stuf...
#</VirtualHost>

# Experiment with authorization...
#<Directory /u/ben/www/1/docs>
#AuthType Basic
#AuthName Experimental
#AuthGroupFile /dev/null
#AuthUserFile /u/ben/www/1/users
```

```
#<Limit PUT GET>
#allow from all
#require valid-user
#</Limit>
#</Directory>

#ScriptAlias /scripts /u/ben/www/scripts

#<VirtualHost ServerName:443>
#SSLEnable
#</VirtualHost>
```

.3 Fichier prelude-manager.conf

```
[Prelude Manager]

# Address where the sensors serber is listening on.
# if value is 127.0.0.1 (or is resolved as being 127.0.0.1),
# it mean the Manager server will be listening via a local (UNIX)
# socket.
#
# format : address:port
#
sensors-srvr = 192.168.0.2;


# Address where the administrative server is listening on.
# if value is "unix", it mean the report server is listening
# on the same machine via a local (UNIX) socket.
#
# format : address:port
#
# admin-srvr = 0.0.0.0:5555;


# If you want the message caught by this manager to be relayed.
# You can use boolean AND and OR to make the rule.
#
# relay-manager = x.x.x.x || y.y.y.y && z.z.z.z
#
# This mean the emission should occur on x.x.x.x or, if it fail,
# on y.y.y.y and z.z.z.z (if one of the two host in the AND fail,
# the emission will be considered as failed involving saving the
# message locally).


#####################################
# Here start plugins configuration #
#####################################
```

```
# [MySQL]

# Host the database is listening on.
dbhost = localhost;

# Name of the database.
dbname = prelude;

# Username to be used to connect the database.
dbuser = prelude;

# Password used to connect the database.
dbpass = dessstri;



#
# The Textmod plugin allow to report alert as text
# in a file. Or to dump theses alert to stderr.
#
# The default logfile for this plugin is /var/log/prelude.log
#

[TextMod]
#
# Tell Textmod to output to stderr
# stderr;
#

logfile = /var/log/prelude.log;



#
# The Xmlmod plugin allow to report alert as IDMEF XML
# in a file. Or to dump theses alert to stderr.
#
# The default logfile for this plugin is /var/log/prelude-xml.log
#

[XmlMod]
#
# Tell Xmlmod to output to stderr
# stderr;
#
# Tell Xmlmod to check generated XML against IDMEF DTD
# check-dtd;
#
```

```
logfile = /var/log/prelude-xml.log;



# [Debug]
#
# Print the value of each element.
# verbose;
#
# Be aggressive, print strings even if consistency checks fail
# (may lead to crash).
# aggressive;
#
# Use wide format for lists.
# wide-format;
```

.4 Fichier prelude-nids.conf

```
################################################
# Configuration for the Prelude NIDS Sensor  #
################################################

[Prelude NIDS]

# Address where the Prelude Manager Server is listening on.
# if value is "127.0.0.1", the connection will occur throught
# an UNIX socket.
#
# This entry is disabled. The default is to use the entry
# located in sensors-default.conf... You may overwrite the
# default address for this sensor by uncommenting this entry.
#
 manager-addr = 192.168.0.2:5554;


# Set this entry if you want Prelude NIDS to use a specific user.
#
# user = prelude;



#[Tcp-Reasm]

#
# TCP stream reassembly option
#
# Only analyse TCP packet that are part of a stream,
```

```
# this defeat stick/snot against TCP signatures.
#
# statefull-only;

#
# Only reassemble TCP data sent by the client (default).
#
# client-only;

#
# Only reassemble TCP data sent by the server.
#
# server-only;

#
# Reassemble TCP data sent by client and server.
#
# both;

#
# Don't reassemble data until we queued a minimum of byte (default is 8192).
#
# min-length = 8192;

#
# Only reassemble data to specific port (default is to reassemble everything)
#
# If this option is used with the statefull-only option, packet that are not
# going to theses specified port will be analyzed anyway.
#
# port-list = 1 2 3 4;


#####################################
# Here start plugins configuration #
#####################################

[SnortRules]

ruleset=/usr/local/etc/prelude-nids/ruleset/prelude.rules;


[ScanDetect]

# Number of connection attempt to get from the same
# host and targeted on different port before the scan
# detection plugin issue an alert.
#
high-port-cnx-count = 50;
low-port-cnx-count = 5;
```

```
# Window of time without getting any activity the scan
# detection plugin should wait before issuing an alert
# for a given host.
#
cnx-ttl = 60;



# [Shellcode]
#
# This plugin allow for polymorphic shellcode detection.
# It may consume a lot of CPU time, so it’s disabled by
# default. Uncomment the section name to enable it, or
# specify --shellcode on the command line.

nops_raise_alert = 60;

#
# If a port-list is specified, the Shellcode plugin
# will only analyse data going to theses port (when
# the protocol used have have dst port).
#
# port-list = 1 2 3 4;



# [Debug]
#
# This plugin issue an alert for each packet.
# Carefull to the loging activity it generate.


[HttpMod]
#
# Normalize HTTP request.
# The "codepage-file" option contains the name of the file containing
# Unicode to ASCII convertion tables for WIN32 machines.
#
# The "codepage-number" option is the codepage number your WIN32 servers use.
#
#
# end-on-param:
# Stop parsing the URL when we meet a parameter.
#
# double-encode:
# Check for encoded ’%’ character.
#
# max-whitespace:
```

```
# Maximum number of whitespace allowed before URL begin.
#
# flip-backslash:
# Change '\' to '/' when parsing URL.
#

double-encode;
flip-backslash;
max-whitespace = 10;
codepage-file = /usr/local/etc/prelude-nids/unitable.txt;
codepage-number = 437;


port-list = 80 8080;


[RpcMod]
#
# Decode RPC traffic, Also provide the RPC rule key.
#
port-list = 111;


[TelnetMod]
#
# Normalize telnet negotiation character
#
port-list = 23 21;


[ArpSpoof]
#
# Search anomaly in ARP request.
#
# The "directed" option will result in a warn each time an ARP
# request is sent to an address other than the broadcast address.
#
# directed;
# arpwatch=<ip> <macaddr>;
```

.5 Fichier prelude-lml.conf

```
##############################################
# Configuration for the Prelude LML Sensor   #
##############################################

[Prelude LML]
```

```
# Address where the Prelude Manager Server is listening on.
# if value is "127.0.0.1", the connection will occur throught
# an UNIX socket.
#
# This entry is disabled. The default is to use the entry
# located in sensors-default.conf... You may overwrite the
# default address for this sensor by uncommenting this entry.
#
 manager-addr = 192.168.0.2;


# Configuration for the UDP message receiver.
# commented out by default since most people only want to
# monitor files.
#
# [Udp-Srvr]
#
# port = 514
# addr = 0.0.0.0



#
# Files to monitor
#
file = /var/log/auth.log
file = /var/log/messages


#####################################
# Here start plugins configuration #
#####################################

[SimpleMod]

ruleset=/usr/local/etc/prelude-lml/ruleset/simple.rules;


# [Debug]
#
# This plugin issue an alert for each packet.
# Carefull to the loging activity it generate.
```

.6 Fichier config.pl

```
sub LoadConfig()
```

```
{
    # Database :
  $conf{'dbtype'}='mysql';
  $conf{'dbname'}='prelude';
  $conf{'dbhost'}='localhost';
  $conf{'dbport'}=3306; # default mysql port is 3306
#  $conf{'dboptions'}='mysql_compression=1';
  $conf{'dblogin'}='prelude';
  $conf{'dbpasswd'}='dessstri';

    # Other :
  $conf{'debug'}=1; # Debug perl code onscreen (0 or 1)
  $conf{'extension'}='.pl'; # scripts file extension (.pl)

  $conf{'refresh'}=600; # AlertList refresh in seconds (600)

  $conf{'ettercap_fp_db'}='./generated/DB/etter.passive.os.fp';
  $conf{'ettercap_mac_db'}='./generated/DB/mac-fingerprints';

  $conf{'HostName_Lookup'}=1; # Host Name Resolution (0 or 1)

  $conf{'GD_transparent'}=0; # Are graphs transparent ?
}

1;
```

.7 Fichier config.php

```
<?
/*
 *
 * Copyright (C) 2002 Vergoz Michael <descript@sysdoor.net>
 * All Rights Reserved
 *
 * This file is part of the Prelude program.
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2, or (at your option)
 * any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
```

```
 * along with this program; see the file COPYING.  If not, write to
 * the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
 *
 */
?>

<?

$server[0]['description'] = "SYSDOOR/PostGreSQL phpfront v".VERSION;
$server[0]['dbtype'] = USE_DB_PGSQL;
$server[0]['dbusername'] = "";
$server[0]['dbpassword'] = "";
$server[0]['dbhostname'] = LOCAL_CONNECTION;
$server[0]['dbport'] = DEFAULT_PORT;
$server[0]['dbname'] = "prelude";

$server[1]['description']       =       "SYSDOOR/MySQL phpfront v".VERSION;
$server[1]['dbtype']            =       USE_DB_MYSQL;
$server[1]['dbusername']        =       "prelude";
$server[1]['dbpassword']        =       "dessstri";
$server[1]['dbhostname']        =       LOCAL_CONNECTION;
$server[1]['dbport']            =       DEFAULT_PORT;
$server[1]['dbname']            =       "prelude";


/*
 * Local variables:
 * tab-width: 4
 * c-basic-offset: 4
 * End:
 * vim600: noet sw=4 ts=4 fdm=marker
 * vim<600: noet sw=4 ts=4
 */
?>
```